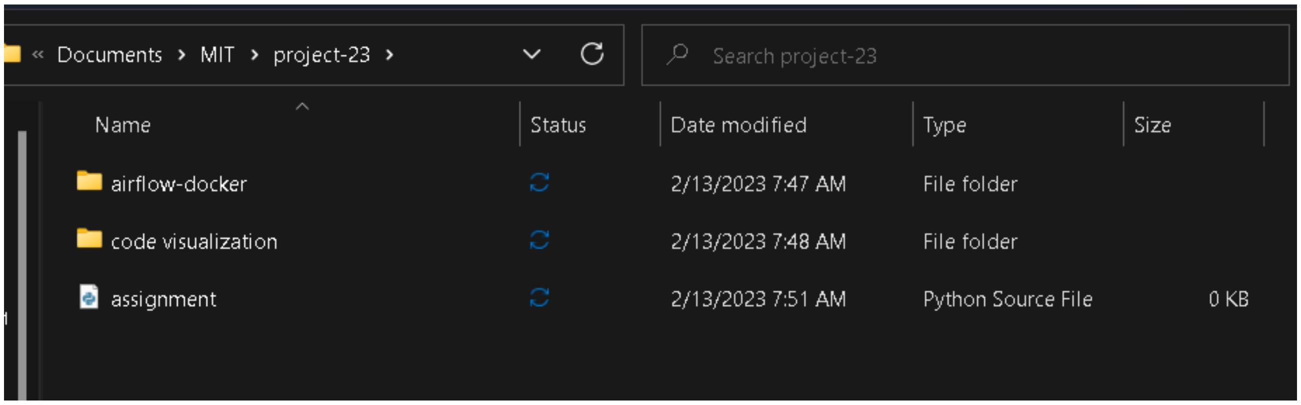
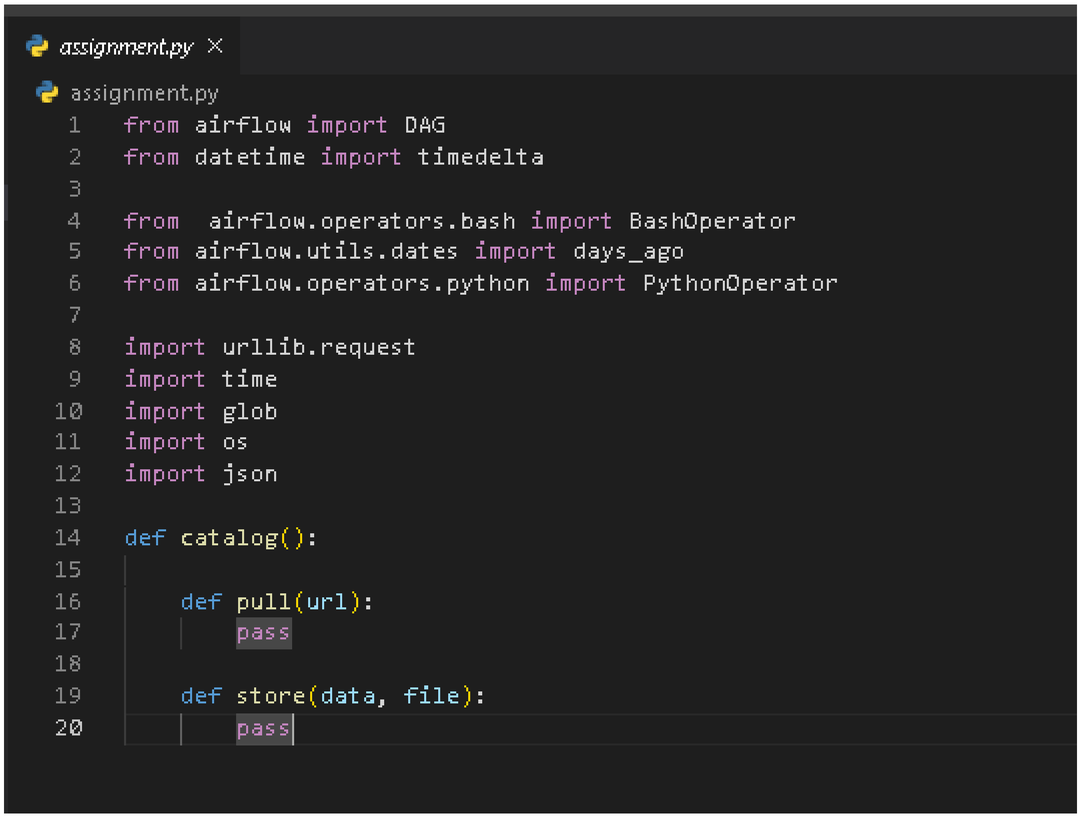
1


« Documents › MIT › project-23 ›

Search project-23

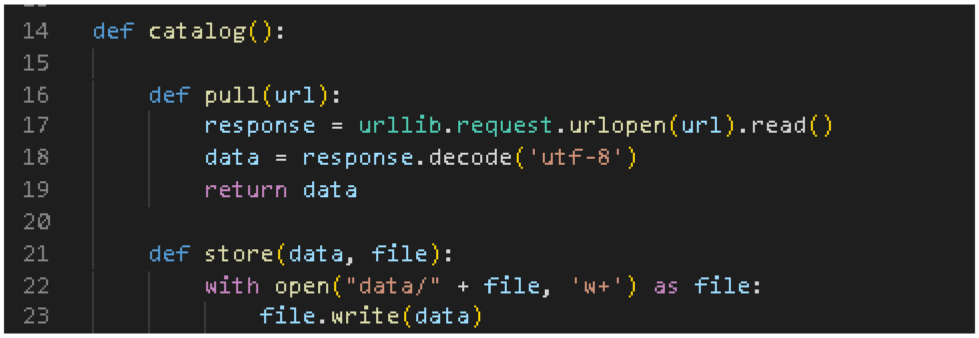
Name	Status	Date modified	Type	Size
airflow-docker		2/13/2023 7:47 AM	File folder	
code visualization		2/13/2023 7:48 AM	File folder	
assignment		2/13/2023 7:51 AM	Python Source File	0 KB

2

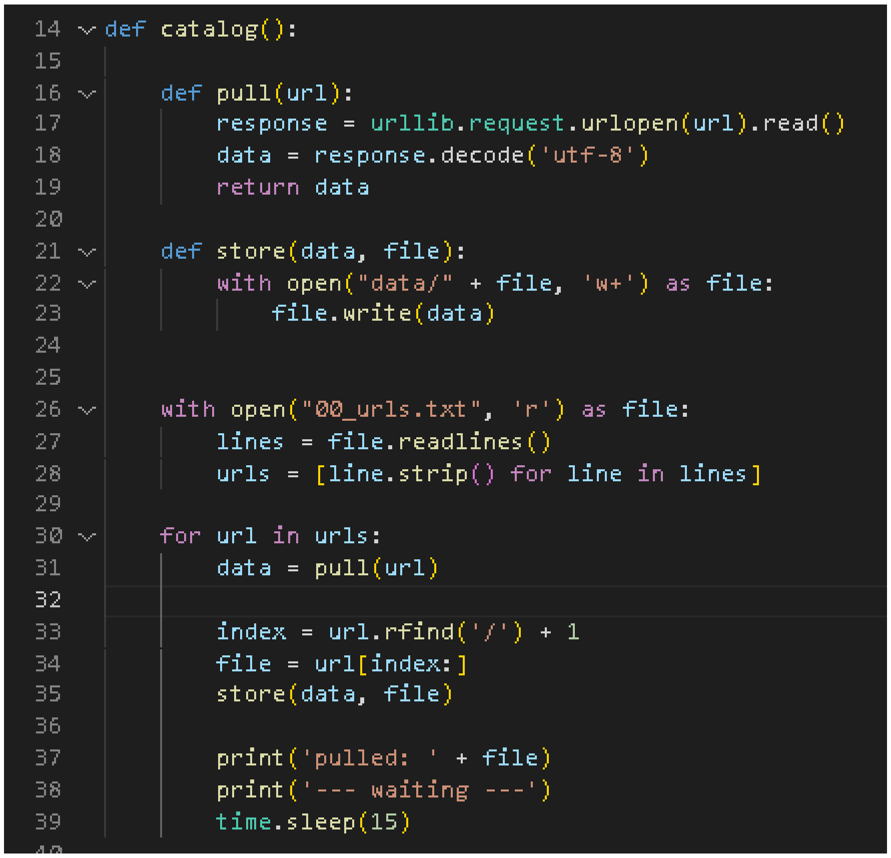

assignment.py

```python
from airflow import DAG
from datetime import timedelta

from  airflow.operators.bash import BashOperator
from airflow.utils.dates import days_ago
from airflow.operators.python import PythonOperator

import urllib.request
import time
import glob
import os
import json

def catalog():

    def pull(url):
        pass

    def store(data, file):
        pass
```

3

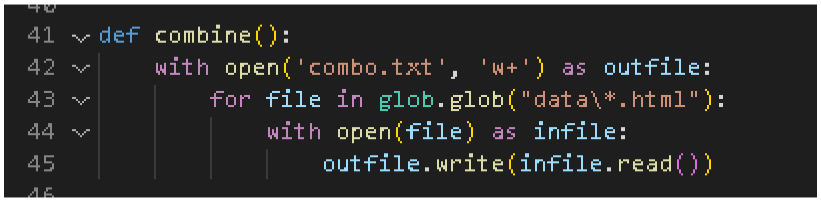

```python
def catalog():

    def pull(url):
        response = urllib.request.urlopen(url).read()
        data = response.decode('utf-8')
        return data

    def store(data, file):
        with open("data/" + file, 'w+') as file:
            file.write(data)
```

4



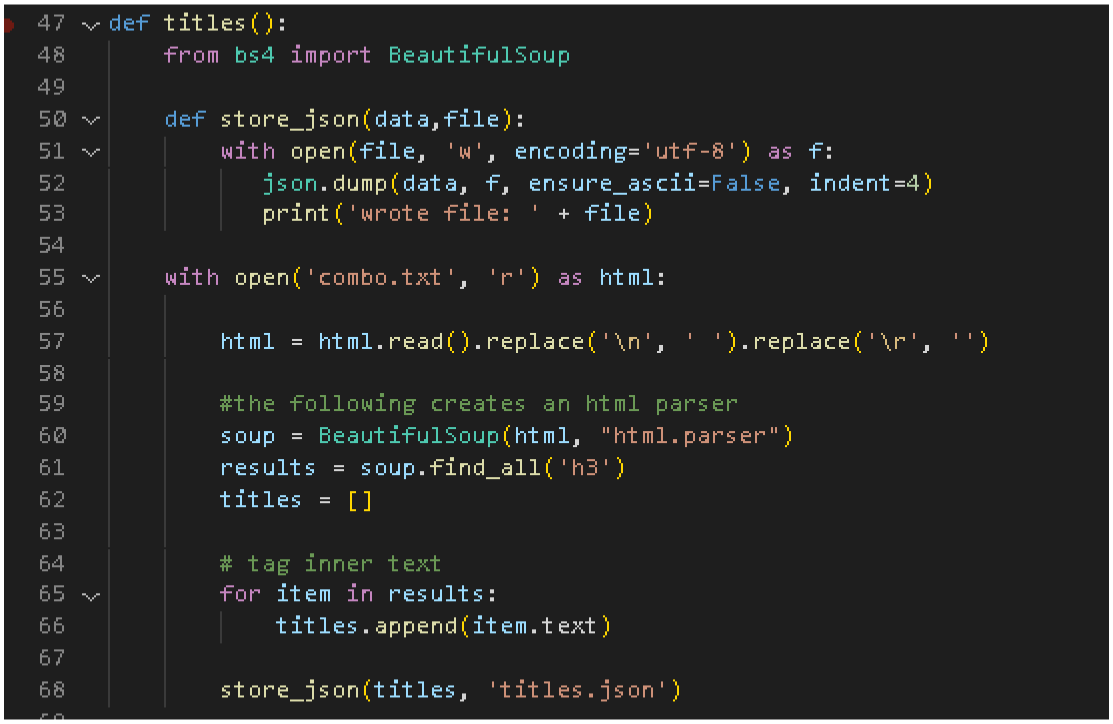
```python
def catalog():

    def pull(url):
        response = urllib.request.urlopen(url).read()
        data = response.decode('utf-8')
        return data

    def store(data, file):
        with open("data/" + file, 'w+') as file:
            file.write(data)


    with open("00_urls.txt", 'r') as file:
        lines = file.readlines()
        urls = [line.strip() for line in lines]

    for url in urls:
        data = pull(url)

        index = url.rfind('/') + 1
        file = url[index:]
        store(data, file)

        print('pulled: ' + file)
        print('--- waiting ---')
        time.sleep(15)
```

5



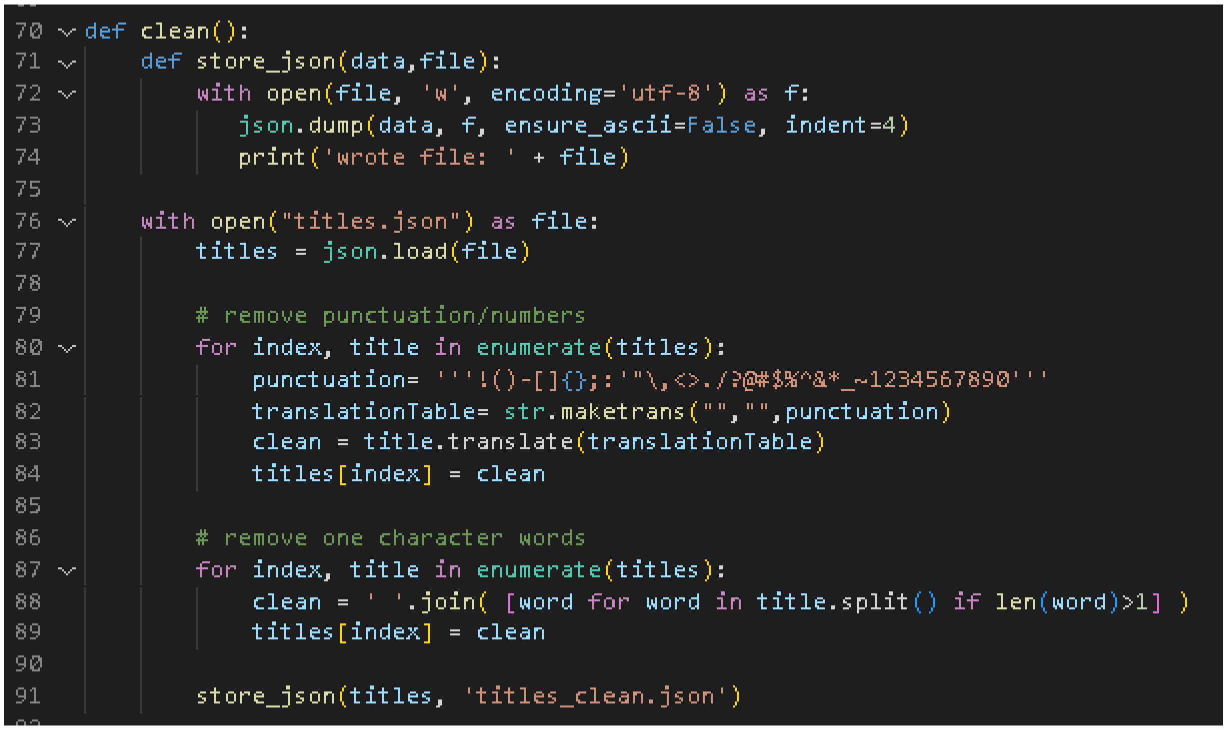
```python
def combine():
    with open('combo.txt', 'w+') as outfile:
        for file in glob.glob("data\*.html"):
            with open(file) as infile:
                outfile.write(infile.read())
```

6



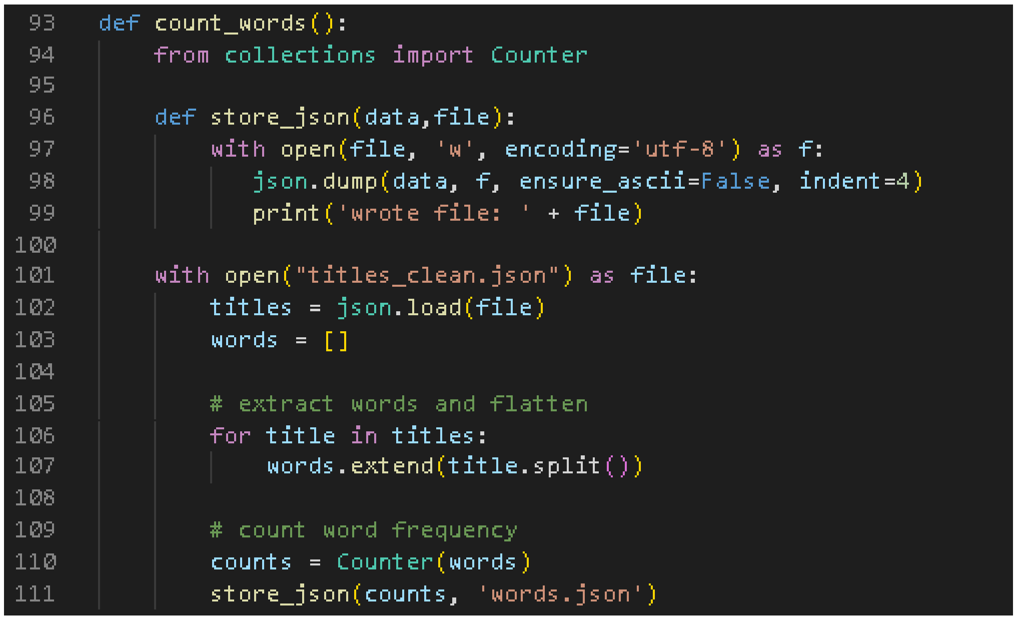
```python
def titles():
    from bs4 import BeautifulSoup

    def store_json(data,file):
        with open(file, 'w', encoding='utf-8') as f:
            json.dump(data, f, ensure_ascii=False, indent=4)
            print('wrote file: ' + file)

    with open('combo.txt', 'r') as html:

        html = html.read().replace('\n', ' ').replace('\r', '')

        #the following creates an html parser
        soup = BeautifulSoup(html, "html.parser")
        results = soup.find_all('h3')
        titles = []

        # tag inner text
        for item in results:
            titles.append(item.text)

        store_json(titles, 'titles.json')
```

7

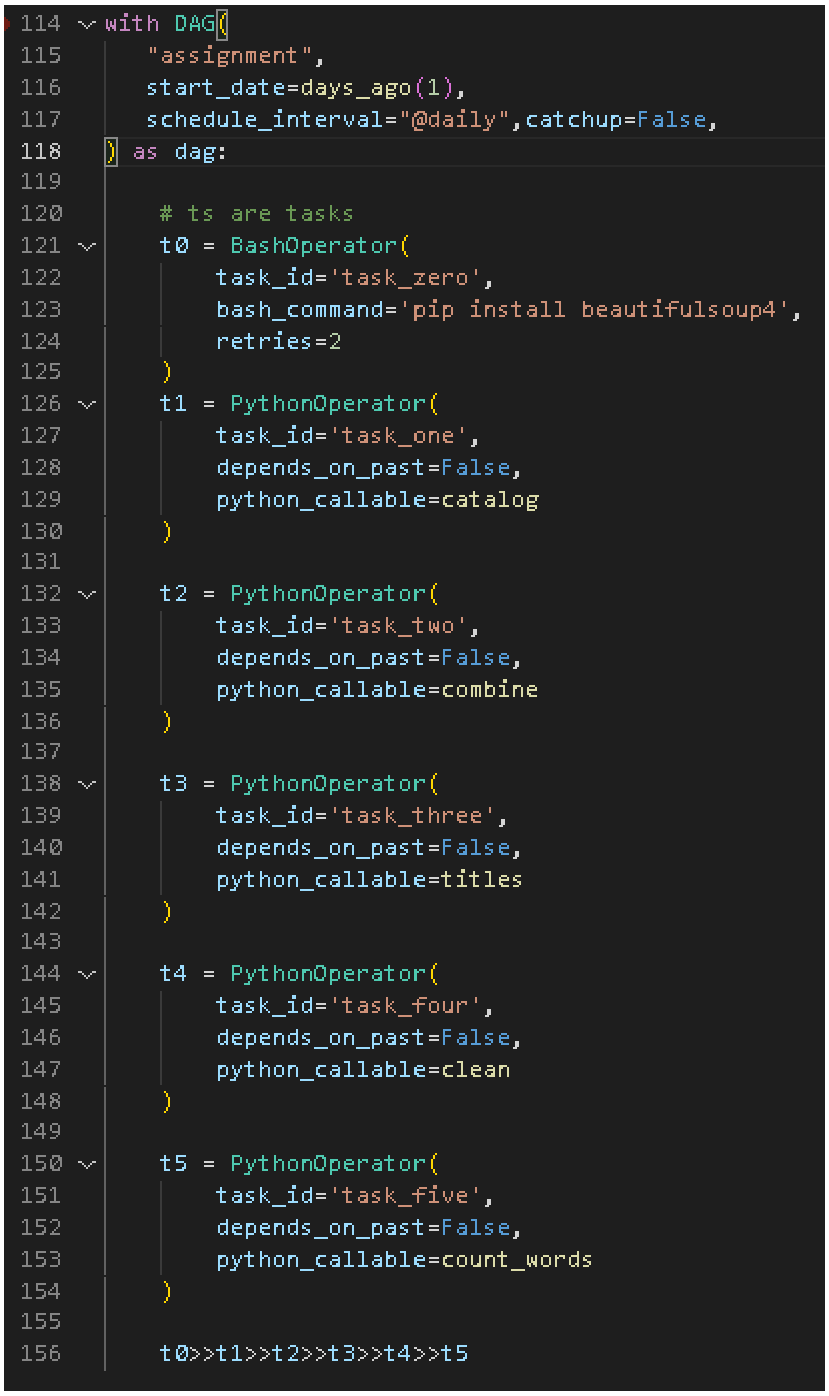


```python
def clean():
    def store_json(data,file):
        with open(file, 'w', encoding='utf-8') as f:
            json.dump(data, f, ensure_ascii=False, indent=4)
            print('wrote file: ' + file)

    with open("titles.json") as file:
        titles = json.load(file)

        # remove punctuation/numbers
        for index, title in enumerate(titles):
            punctuation= '''!()-[]{};:'"\,<>./?@#$%^&*_~1234567890'''
            translationTable= str.maketrans("","",punctuation)
            clean = title.translate(translationTable)
            titles[index] = clean

        # remove one character words
        for index, title in enumerate(titles):
            clean = ' '.join( [word for word in title.split() if len(word)>1] )
            titles[index] = clean

        store_json(titles, 'titles_clean.json')
```

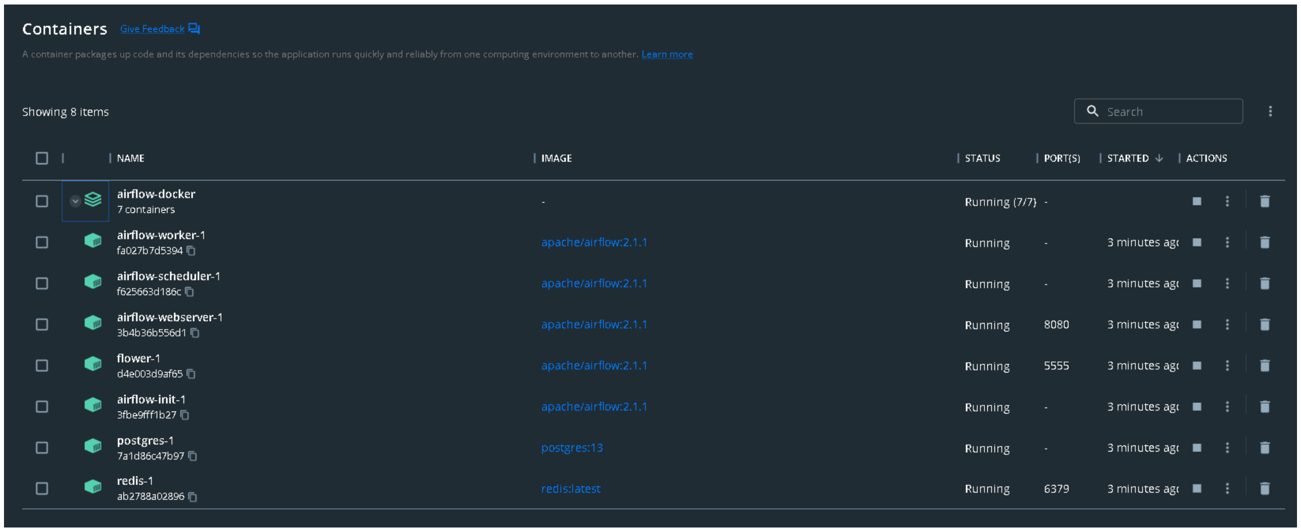
8



```python
def count_words():
    from collections import Counter

    def store_json(data,file):
        with open(file, 'w', encoding='utf-8') as f:
            json.dump(data, f, ensure_ascii=False, indent=4)
            print('wrote file: ' + file)

    with open("titles_clean.json") as file:
        titles = json.load(file)
        words = []

        # extract words and flatten
        for title in titles:
            words.extend(title.split())

        # count word frequency
        counts = Counter(words)
        store_json(counts, 'words.json')
```



```python
with DAG(
    "assignment",
    start_date=days_ago(1),
    schedule_interval="@daily",catchup=False,
) as dag:

    # ts are tasks
    t0 = BashOperator(
        task_id='task_zero',
        bash_command='pip install beautifulsoup4',
        retries=2
    )
    t1 = PythonOperator(
        task_id='task_one',
        depends_on_past=False,
        python_callable=catalog
    )

    t2 = PythonOperator(
        task_id='task_two',
        depends_on_past=False,
        python_callable=combine
    )

    t3 = PythonOperator(
        task_id='task_three',
        depends_on_past=False,
        python_callable=titles
    )

    t4 = PythonOperator(
        task_id='task_four',
        depends_on_past=False,
        python_callable=clean
    )

    t5 = PythonOperator(
        task_id='task_five',
        depends_on_past=False,
        python_callable=count_words
    )

    t0>>t1>>t2>>t3>>t4>>t5
```



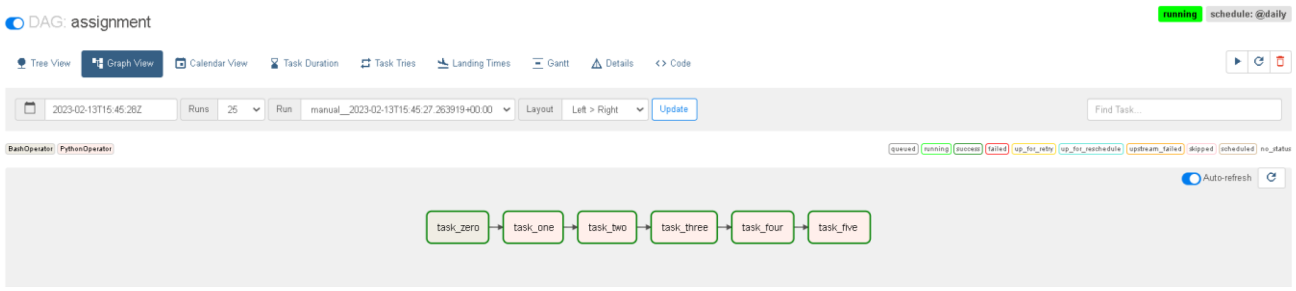
Containers Give Feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Showing 8 items

NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
airflow-docker 7 containers	-	Running (7/7)	-		
airflow-worker-1 fa027b7d5394	apache/airflow:2.1.1	Running	-	3 minutes ago	
airflow-scheduler-1 f625663d186c	apache/airflow:2.1.1	Running	-	3 minutes ago	
airflow-webserver-1 3b4b36b556d1	apache/airflow:2.1.1	Running	8080	3 minutes ago	
flower-1 d4e003d9af65	apache/airflow:2.1.1	Running	5555	3 minutes ago	
airflow-init-1 3fbe9fff1b27	apache/airflow:2.1.1	Running	-	3 minutes ago	
postgres-1 7a1d86c47b97	postgres:13	Running	-	3 minutes ago	
redis-1 ab2788a02896	redis:latest	Running	6379	3 minutes ago	

11



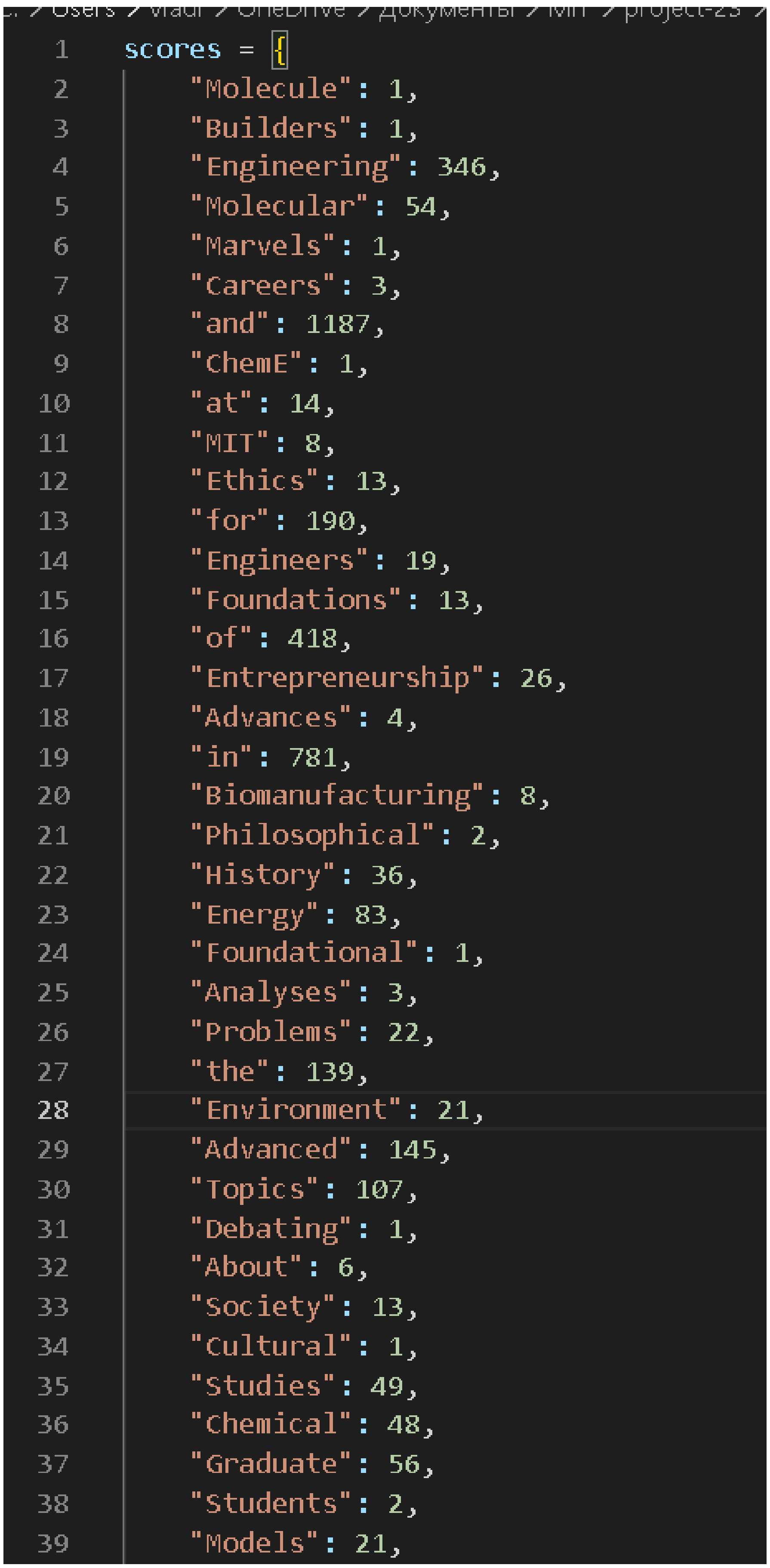
DAG: assignment

running schedule: @daily

Run: manual__2023-02-13T15:45:27.263919+00:00

2023-02-13T15:45:28Z Runs 25 Layout Left > Right

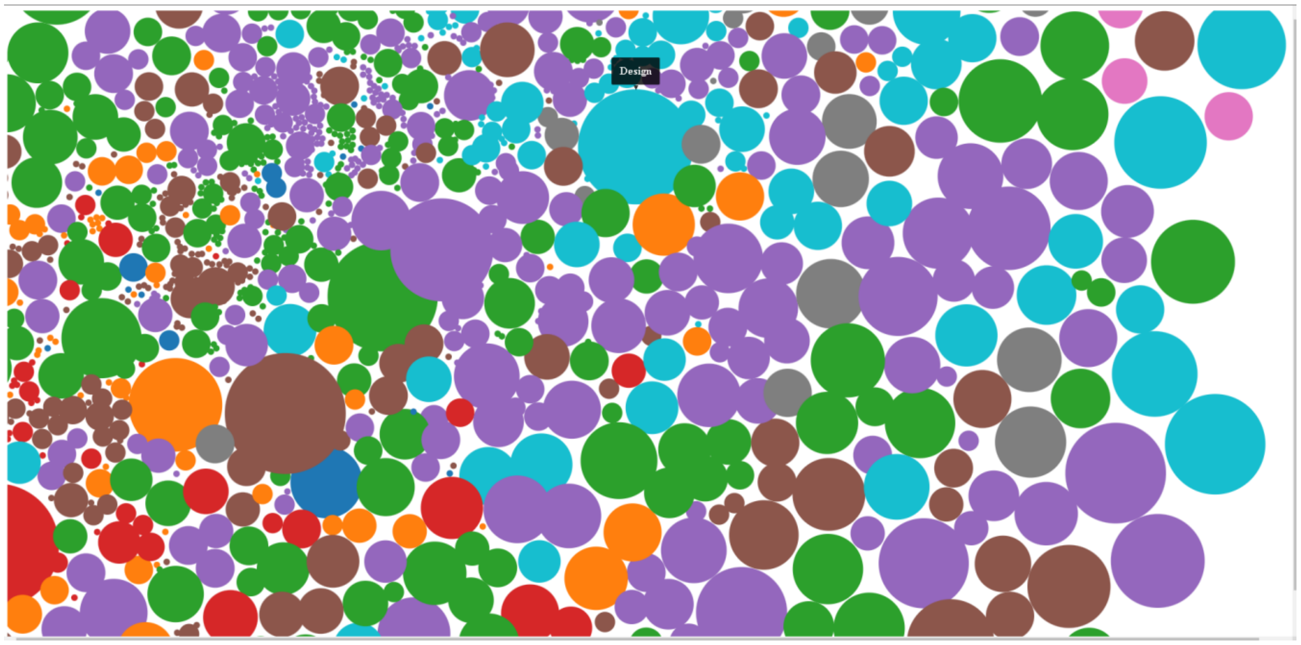
12



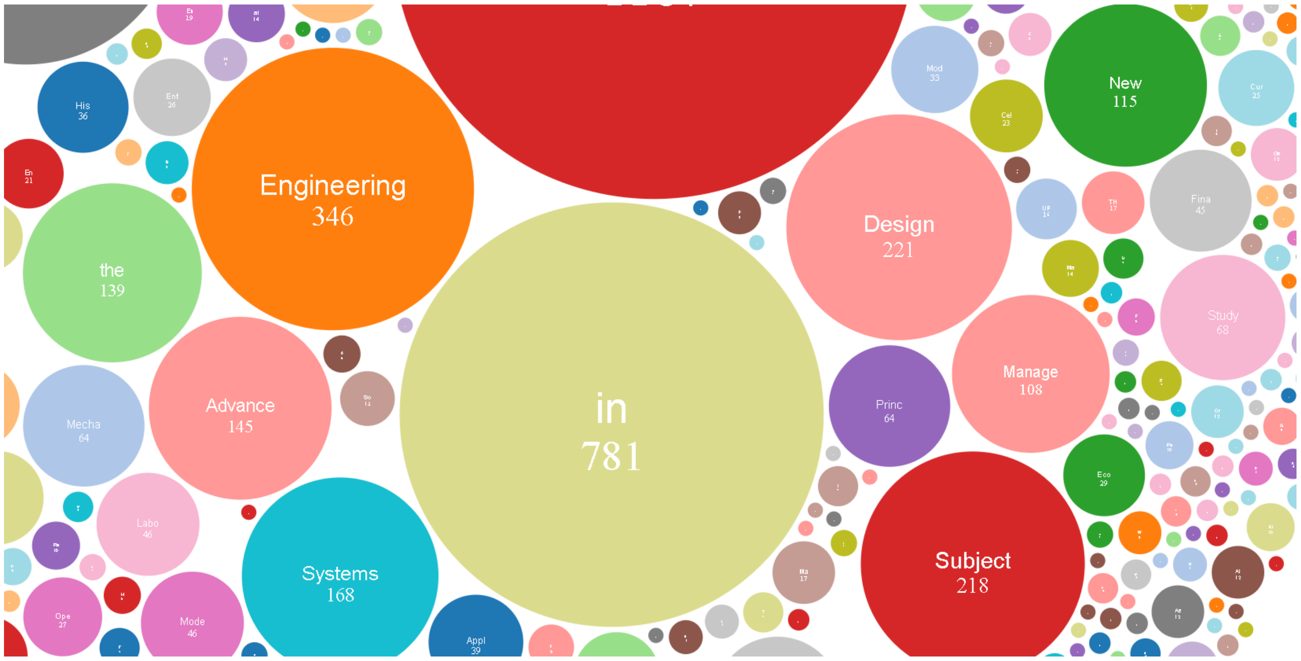
C: > Users > vladi > OneDrive > Документы > MIT > project-23 >

```python
scores = {
    "Molecule": 1,
    "Builders": 1,
    "Engineering": 346,
    "Molecular": 54,
    "Marvels": 1,
    "Careers": 3,
    "and": 1187,
    "ChemE": 1,
    "at": 14,
    "MIT": 8,
    "Ethics": 13,
    "for": 190,
    "Engineers": 19,
    "Foundations": 13,
    "of": 418,
    "Entrepreneurship": 26,
    "Advances": 4,
    "in": 781,
    "Biomanufacturing": 8,
    "Philosophical": 2,
    "History": 36,
    "Energy": 83,
    "Foundational": 1,
    "Analyses": 3,
    "Problems": 22,
    "the": 139,
    "Environment": 21,
    "Advanced": 145,
    "Topics": 107,
    "Debating": 1,
    "About": 6,
    "Society": 13,
    "Cultural": 1,
    "Studies": 49,
    "Chemical": 48,
    "Graduate": 56,
    "Students": 2,
    "Models": 21,
```

13


Design

14


Engineering
346
the
139
in
781
Design
221
New
115
Manage
108
Subject
218
Systems
168
Advance
145
Mecha
64
Princ
64
Study
68
Fina
45
Labo
46
Mode
46
Appl
39
His
36
Mod
33
Ent
26
Eco
29
Ope
27
En
21
Cur
25
Col
23