4-io

February 2, 2024

1 Python Programming (Basic-Intermediate)

1.1 Module 4 - IO

1.2 File operation

[14]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[ ]:

```
f = open('/content/drive/MyDrive/AIS_DG/gpl-3.0.txt')
content = f.read()
f.close()
print(content)
```

[ ]:

```
f.closed
```

[ ]:

```
f.mode
```

[ ]:

```
f.name
```

[ ]:

```
f1 = open('/content/drive/MyDrive/AIS_DG/superstore.data', mode='rb')
x = f1.read()
f1.close()
x
```

[ ]:

```
with open('/content/drive/MyDrive/AIS_DG/gpl-3.0.txt') as f:
  content = f.read()
  print(content)
```

1.3 Reading line by line

[ ]:

```
filename = '/content/drive/MyDrive/AIS_DG/gpl-3.0.txt'

with open(filename) as file_object:
  ind = 1
  for line in file_object:
    print(ind,' ', line)
    ind += 1
```

1.4 Reading a list of lines from a file

[ ]:

```
filename = '/content/drive/MyDrive/AIS_DG/gpl-3.0.txt'

with open(filename) as file_object:
  contents = file_object.readlines()

print('Number of Lines: ', len(contents))
print(contents)
```

1.5 Create and write a file

[ ]:

```
with open('/content/drive/MyDrive/AIS_DG/programming.txt','w') as f:
  f.write('I love programming.')
```

[ ]:

```
!cat /content/drive/MyDrive/AIS_DG/programming.txt
```

1.6 File position

[ ]:

```
pos = 0
with open('/content/drive/MyDrive/AIS_DG/gpl-3.0.txt') as f:
  print(f.readline())
  pos = f.tell()
  print(pos, ' ', f.readline())
  pos = f.tell()
  print(pos, ' ', f.readline())
  pos = f.tell()
  print(pos, ' ', f.readline())
  print('=== Seek to 47 from the beginning ===')
  f.seek(47, 0)
  pos = f.tell()
  print(pos, ' ', f.readline())
```

1.7 os package

[ ]:
```
import os
os.listdir('/content/drive/MyDrive/AIS_DG/')
```

[ ]:
```
os.rename('/content/drive/MyDrive/AIS_DG/programming.txt',
          '/content/drive/MyDrive/AIS_DG/p1.txt')
os.listdir('/content/drive/MyDrive/AIS_DG/')
```

[ ]:
```
import glob
glob.glob('/content/drive/MyDrive/AIS_DG/*.csv')
```

[ ]:
```
os.environ
```

1.8 Reading CSV (Pandas DataFrame)

[ ]:
```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/AIS_DG/Telco-Churn.csv')
df.info()
```

[ ]:
```
df_chunks = pd.read_csv('/content/drive/MyDrive/AIS_DG/Telco-Churn.csv',
                        iterator=True, chunksize=700)
```

[ ]:
```
for d in df_chunks:
    print(max(d.index))
```

[ ]:
```
x1 = df_chunks.get_chunk()
print(x1.shape)
x1.head()
```

[ ]:
```
x2 = df_chunks.get_chunk()
print(x2.shape)
x2.head()
```

[ ]:
```
df_chunks
```

1.9 Reading an Excel file

[ ]:
```
superstore = pd.read_excel('/content/drive/MyDrive/AIS_DG/Superstore.xlsx')
superstore.head()
```

[ ]:
```
superstore_file = pd.ExcelFile('/content/drive/MyDrive/AIS_DG/Superstore.xlsx')
```

[ ]:
```
sn = superstore_file.sheet_names
```

[ ]:
```
d_excel = pd.read_excel(superstore_file, sheet_name=sn)
```

[ ]:
```
d_excel['People']
```

1.10 Reading JSON

[ ]:
```
import json

f = open('/content/drive/MyDrive/AIS_DG/btc.json')

content = json.load(f)

f.close()
print(content)
```

[ ]:
```

```

[ ]:
```
d = json.load(open('/content/drive/MyDrive/AIS_DG/cc.json'))
d[:2]
```

[ ]:
```
pd.read_json('/content/drive/MyDrive/AIS_DG/cc.json')
```

1.11 Read/write pickles

[ ]:
```
import glob

superstore.to_pickle('/content/drive/MyDrive/AIS_DG/superstore.data')
print(glob.glob('/content/drive/MyDrive/AIS_DG/*.data'))
```

[ ]:
```
df1 = pd.read_pickle('/content/drive/MyDrive/AIS_DG/superstore.data')
df1.head()
```

[ ]:
```
import pickle
```

[ ]:
```
df1.to_pickle('test.data')
```

1.12 Read file from URL

[ ]:
```
weather = pd.read_csv('http://fastdata.in.th/AIS/weather_daily_darksky.csv')
weather.head()
```

1.13 API call

[ ]:
```
import requests

resp = requests.get('https://api.coingecko.com/api/v3/coins/markets/?
  ↪vs_currency=usd')
print(resp.status_code)
```

```
crypto_data = resp.json()
print(crypto_data)
```

1.14 Activity

Add a function to myutils.py

Function Name: load_current_weather

Description: extract data from TMD Weather API and append it to file specified in the argument. If not exist, create a new file.

Import and test the function.

TMD Weather URL: https://data.tmd.go.th

[1]:

```
# work here
%%writefile myutils.py


def load_current_weather(dest:str):
  """ load and extract data from api to dataframe format and save
      arg:
        dest: file path
  """
  import requests
  from bs4 import BeautifulSoup
  import pandas as pd

  # load data from api
  url = "https://data.tmd.go.th/api/Weather3Hours/V2/?uid=api&ukey=api12345"
  response = requests.request("GET", url)
  data = response.text

  # extract data
  soup = BeautifulSoup(data, 'xml')
  list_stations = soup.find_all('Station')

  dt1 = [s.find('DateTime').text for s in list_stations]
  sn1 = [s.find('StationNameThai').text for s in list_stations]
  at1 = [s.find('AirTemperature').text for s in list_stations]

  d = pd.DataFrame({'Date':dt1,'StationName': sn1, 'AirTemperature': at1})

  d.to_csv(dest,mode = 'a',header=False)
```

```
Writing myutils.py
```

[2]:

```
from myutils import load_current_weather
dest = "./result.csv"
load_current_weather(dest)
```

[3]:

```
import pandas as pd
test_df = pd.read_csv(dest)
test_df.head()
```

[3]:

```
   0  02/02/2024 13:00:00           26.2
0  1  02/02/2024 13:00:00            17.0
1  2  02/02/2024 13:00:00             28.4
2  3  02/02/2024 13:00:00        .  28.7
3  4  02/02/2024 13:00:00           31.0
4  5  02/02/2024 13:00:00                31.0
```

[4]:

```
import requests

url = "https://data.tmd.go.th/api/Weather3Hours/V2/?uid=api&ukey=api12345"

response = requests.request("GET", url)

data = response.text
```

[5]:

```
from bs4 import BeautifulSoup
```

[6]:

```
soup = BeautifulSoup(data, 'xml')
```

[7]:

```
list_stations = soup.find_all('Station')
```

[8]:

```
len(list_stations)
```

[8]:

```
128
```

[9]:

```
list_stations[0]
```

[9]:

```
<Station><WmoStationNumber>48300</WmoStationNumber><StationNameThai>    </
StationNameThai><StationNameEnglish>MAE HONG
SON</StationNameEnglish><Province>    </Province><Latitude Unit="decimal
degree">19.29897</Latitude><Longitude Unit="decimal
degree">97.97578</Longitude><Observation><DateTime>02/02/2024
13:00:00</DateTime><StationPressure
unit="hPa">984.30</StationPressure><MeanSeaLevelPressure
Unit="hPa">1014.57</MeanSeaLevelPressure><AirTemperature
Unit="celsius">26.2</AirTemperature><DewPoint
Unit="celsius">14.1</DewPoint><RelativeHumidity
Unit="%">47</RelativeHumidity><VaporPressure
Unit="mb">16.06</VaporPressure><LandVisibility
```

```
Unit="km">10.00</LandVisibility><WindDirection
Unit="degree">000</WindDirection><WindSpeed Unit="km/h">0.0</WindSpeed><Rainfall
Unit="mm">0.00</Rainfall><Rainfall24Hr
Unit="mm">0.00</Rainfall24Hr></Observation></Station>
```

[10]:
```
list_stations[0].find('StationNameThai').text
```

[10]:
```
'     '
```

[11]:
```
dt1 = [s.find('DateTime').text for s in list_stations]
sn1 = [s.find('StationNameThai').text for s in list_stations]
at1 = [s.find('AirTemperature').text for s in list_stations]
```

[12]:
```
d = pd.DataFrame({'Date':dt1,'StationName': sn1, 'AirTemperature': at1})
d.head()
```

[12]:
```
                  Date    StationName AirTemperature
0  02/02/2024 13:00:00                    26.2
1  02/02/2024 13:00:00                     17.0
2  02/02/2024 13:00:00                      28.4
3  02/02/2024 13:00:00        .           28.7
4  02/02/2024 13:00:00                     31.0
```

[15]:
```
arg = '/content/drive/MyDrive/AIS_DG/result.csv'
d.to_csv(arg,mode = 'a',header=False)
```

[16]:
```
pd.read_csv(arg).shape
```

[16]:
```
(371, 4)
```