Freshers Programs

September 5, 2023

[86]:
```
pwd
```

[86]:
```
'/home/niranjan/Desktop/SecureKloud'
```

0.0.1 Day - 1 Python File Operation

[17]:
```
# 1

text = input('Enter text : ')
file_path = input('Enter file path :')

fp = open(file_path, 'w')
fp.write(text)
fp.close()
```

```
Enter text : hi im from file 1
Enter file path :/home/niranjan/Desktop/SecureKloud/sample.txt
```

[19]:
```
read_file = open(file_path, 'r')
print(read_file.read())
```

```
hi im from file 1
```

[20]:
```
f = open('/home/niranjan/Desktop/SecureKloud/sample2.txt', 'w')
f.write('I am from file 2')
f.close()
```

[21]:
```
f2 = open('/home/niranjan/Desktop/SecureKloud/sample3.txt', 'w')
f2.write('I am from file 3')
f2.close()
```

[22]:
```
f3 = open('/home/niranjan/Desktop/SecureKloud/sample4.txt', 'w')
f3.write('I am from file 4')
f3.close()
```

[23]:
```
# 2
```

```
import os

files = []
input_path = input('Enter folder path: ')
for i in os.listdir(input_path):
    if i.endswith('.txt'):
        files.append(i)

content = ''

for j in files:
    fp = open(input_path+j, 'r')
    data = fp.read()
    data = data+'\n'
    content+=data


output_file = open('/home/niranjan/Desktop/SecureKloud/output_file.txt', 'w')
output_file.write(content)
output_file.close()

read_output = open('/home/niranjan/Desktop/SecureKloud/output_file.txt', 'r')
print(read_output.read())
```

```
Enter folder path: /home/niranjan/Desktop/SecureKloud/
I am from file 4
I am from file 4

I am from file 3
Hi I am from file 1
I am from file 2

I am from file 3
hi im from file 1
I am from file 2
```

0.0.2 Day 2 - CSV to JSON, Logging

[53]:
```
# 1

import csv
import json

data = []
input_csv = open('/home/niranjan/Desktop/SecureKloud/emp_details.csv', 'r')
reader = csv.DictReader(input_csv)
```

```
for i in reader:
    data.append(i)

json_file = open('emp_details.json', 'a')
json.dump(data, json_file)
```

[54]:
```
read_json = open('emp_details.json', 'r')
print(read_json.read())
```

```
[{"Name": "Niranjan", "Gender": "m", "Age": "21"}, {"Name": "Mahendar",
"Gender": "m", "Age": "22"}, {"Name": "Vel", "Gender": "m", "Age": "20"},
{"Name": "Jayanth", "Gender": "m", "Age": "20"}]
```

[68]:
```
# 2 - logging functionality

import csv
import logging

logging.basicConfig(filename='my_log2.log', level=logging.DEBUG,
                    format='%(asctime)s - %(levelname)s - %(message)s')

def file_working(file):
    data = open(file, 'r')
    try:
        read_file = open(file, 'r')
        data = csv.reader(read_file)
        for i in data:
            print(i)
        logging.info('File accessed')
    except:
        logging.error('File not opening')

file_path = input('Enter file name: ')
file_working(file_path)

logging.debug('This is a debug message')
logging.info('This is an info message')
logging.warning('This is a warning message')
logging.error('This is an error message')
logging.critical('This is a critical message')
```

```
Enter file name: /home/niranjan/Desktop/SecureKloud/emp_details.csv
['Name', 'Gender', 'Age']
['Niranjan', 'm', '21']
['Mahendar', 'm', '22']
['Vel', 'm', '20']
['Jayanth', 'm', '20']
```

0.0.3 Day 3 - Pandas Framework

[19]:
```
import pandas as pd

data1 = pd.read_csv('/home/niranjan/Desktop/SecureKloud/employee1.csv')
print(data1)
```

```
       name  gender   age
0  niranjan       m    21
1  mahendar       m    22
2       vel       m    21
3   jayanth       m    16
4     mukul       m    15
```

[20]:
```
data2 = pd.read_csv('/home/niranjan/Desktop/SecureKloud/employee2.csv')
print(data2)
```

```
       name   years
0  niranjan       2
1  mahendar       3
2       vel       4
3   jayanth       5
4     mukul       6
```

[21]:
```
data3 = pd.merge(data1, data2,how='inner')
print(data3)
```

```
       name  gender   age   years
0  niranjan       m    21       2
1  mahendar       m    22       3
2       vel       m    21       4
3   jayanth       m    16       5
4     mukul       m    15       6
```

[22]:
```
data3.to_csv('/home/niranjan/Desktop/SecureKloud/EmployeeDetails.csv')
```

[23]:
```
df = pd.read_csv('/home/niranjan/Desktop/SecureKloud/EmployeeDetails.csv',␣
↪index_col=0)
print(df)
```

```
       name  gender   age   years
0  niranjan       m    21       2
1  mahendar       m    22       3
2       vel       m    21       4
3   jayanth       m    16       5
4     mukul       m    15       6
```

[24]:
```
# 2

emp_data = pd.read_csv('/home/niranjan/Desktop/SecureKloud/EmployeeDetails.csv')
emp_performance = pd.read_csv('/home/niranjan/Desktop/SecureKloud/
 ↪emp_performance.csv')
print(emp_performance)
```

```
       name   january   feb   march   april
0  niranjan         6   8.0       9        10
1  mahendar         7   7.5       8         9
2       vel         8   8.0       9         9
3   jayanth         9  10.0       6         7
4     mukul        10   6.0       7         9
```

[25]:
```
avg_performance = []
for i in range(0, len(emp_performance['name'])):
    mean = round(emp_performance.iloc[i,1:].mean(), 2)
    avg_performance.append(mean)

emp_performance['avg performance'] = avg_performance
print(emp_performance)
```

```
       name   january   feb   march   april    avg performance
0  niranjan         6   8.0       9        10             8.25
1  mahendar         7   7.5       8         9             7.88
2       vel         8   8.0       9         9             8.50
3   jayanth         9  10.0       6         7             8.00
4     mukul        10   6.0       7         9             8.00
```

[26]:
```
import pandas as pd

output_file = pd.DataFrame({'Employee': emp_performance['name'], 'Avg␣
 ↪Performance': emp_performance['avg performance']})
print(output_file)
output_file.to_csv('emp_avg_performance')
```

```
   Employee  Avg Performance
0  niranjan             8.25
1  mahendar             7.88
2       vel             8.50
3   jayanth             8.00
4     mukul             8.00
```

0.0.4 Day - 4 Pandas Frameworks, Comprehension List, Aggregation using Lambda

[28]:
```
output_file.describe()
```

[28]:
```
       Avg Performance
count         5.000000
mean          8.126000
std           0.248757
min           7.880000
25%           8.000000
50%           8.000000
75%           8.250000
max           8.500000
```

[29]:
```
filtered_data = output_file.loc[output_file['Avg Performance'] > 8, 'Employee']
# filter
print(filtered_data)
```

```
0    niranjan
2         vel
Name: Employee, dtype: object
```

[30]:
```
print(data3)
```

```
      name  gender   age   years
0  niranjan       m    21       2
1  mahendar       m    22       3
2       vel       m    21       4
3   jayanth       m    16       5
4     mukul       m    15       6
```

[32]:
```
data3.loc[:, 'name']
```

[32]:
```
0    niranjan
1    mahendar
2         vel
3     jayanth
4       mukul
Name: name, dtype: object
```

[204]:
```
data = pd.read_csv('EmployeeDetails.csv')
print(data)
```

```
      name  gender   age   years
0  niranjan       m    21       2
1  mahendar       m    22       3
2       vel       m    21       4
3   jayanth       m    16       5
4     mukul       m    15       6
```

[205]:
```
data['name'].count()
```

[205]: 5

[206]:
```
data.shape
```

[206]: (5, 4)

[207]:
```
print(data.isnull().sum())  # no null values
```

```
name       0
 gender    0
 age       0
 years     0
dtype: int64
```

[151]:
```
output_file['Avg Performance'].sum()
```

[151]: 40.629999999999995

[224]:
```
# comprehension list and lambda

fruits = ['apple', 'banana', 'orange', 'watermelon', 'mango']
f = [x for x in fruits if len(x)!=5]
print(f)
```

```
['banana', 'orange', 'watermelon']
```

[237]:
```
b=[1,2,3,4,5]
b2 = [i*i for i in b]
print(b2)
```

```
[1, 4, 9, 16, 25]
```

[235]:
```
def square(L):
    L2 = lambda L : L*L
    result = L2(L)
    print(result)

a = 5
square(a)
```

```
25
```

[43]:
```
import pandas as pd

class Employee():
    def __init__(self, details):
        self.details = details
```

```
    def show_emp(self, name):
        print(self.details[self.details['name'] == name])

    def showall(self):
        print(self.details)

EmployeeDetails = pd.read_csv('EmployeeDetails.csv')

obj = Employee(EmployeeDetails)

obj.show_emp('niranjan')

obj.showall()
```

```
   Unnamed: 0      name  gender   age   years
0           0  niranjan       m    21       2
   Unnamed: 0      name  gender   age   years
0           0  niranjan       m    21       2
1           1  mahendar       m    22       3
2           2       vel       m    21       4
3           3   jayanth       m    16       5
4           4     mukul       m    15       6
```

[55]:
```
import csv

dic = {}

EmployeeDetails = open('EmployeeDetails2.csv', 'r')
data = csv.DictReader(EmployeeDetails)
for i in data:
    dic[i['employee id']] = i.values()

EmployeeDetails.close()

id = input()
print(dic[id])
```

```
1001
dict_values(['1001', 'mahendar', ' m', '22', '3'])
```

[58]:
```
for i in dic:
    print(dic[i])
```

```
dict_values(['1000', 'niranjan', ' m', '21', '2'])
dict_values(['1001', 'mahendar', ' m', '22', '3'])
dict_values(['1002', 'vel', ' m', '21', '4'])
dict_values(['1003', 'jayanth', ' m', '16', '5'])
dict_values(['1004', 'mukul', ' m', '15', '6'])
```

0.0.5 Day - 5 API

[9]:
```
import requests
import csv

api_url = "https://jsonplaceholder.typicode.com/posts"
response = requests.get(api_url)

if response.status_code == 200:
    data = response.json()

    csv_file_path = "api_data.csv"

    with open(csv_file_path, "w", newline="", encoding="utf-8") as csv_file:
        csv_writer = csv.writer(csv_file)
        csv_writer.writerow(["UserId", "Id", "Title", "Body"])

        for item in data:
            csv_writer.writerow([item["userId"], item["id"], item["title"],␣
 ↪item["body"]])
else:
    print("Failed to fetch data from the API.")
```

0.0.6 Day - 6 File Handling and Operations

[11]:
```
from PIL import Image, ImageOps

def merge_images_side_by_side(image1_path, image2_path, output_path):
    image1 = Image.open(image1_path)
    image2 = Image.open(image2_path)

    merged_side_by_side = Image.new('RGB', (image1.width + image2.width,␣
 ↪max(image1.height, image2.height)))
    merged_side_by_side.paste(image1, (0, 0))
    merged_side_by_side.paste(image2, (image1.width, 0))

    merged_side_by_side.save(output_path)

def split_image(image_path, output_path1, output_path2):
    image = Image.open(image_path)
    width, height = image.size

    half_width = width // 2
    left_half = image.crop((0, 0, half_width, height))
    right_half = image.crop((half_width, 0, width, height))

    left_half.save(output_path1)
```

```
    right_half.save(output_path2)

def crop_image(image_path, output_path, x, y, width, height):
    image = Image.open(image_path)
    cropped_image = image.crop((x, y, x + width, y + height))
    cropped_image.save(output_path)

def tilt_page(image_path, output_path, angle):
    image = Image.open(image_path)
    rotated_image = image.rotate(angle, expand=True)
    rotated_image.save(output_path)

merge_images_side_by_side('image1.jpeg', 'image2.jpeg', 'merged_side_by_side.
 ↪jpg')
split_image('road.jpg', 'left_half.jpg', 'right_half.jpg')
crop_image('image1.jpeg', 'cropped_image.jpg', x=100, y=100, width=200,␣
 ↪height=200)
tilt_page('image1.jpeg', 'tilted_page.jpg', angle=30)
```

[18]:
```
# pip install pdf2image

from pdf2image import convert_from_path

def convert_pdf_to_jpeg(pdf_path, output_folder):
    images = convert_from_path(pdf_path)
    for i, image in enumerate(images):
        image.save(f'{output_folder}/page_{i + 1}.jpg', 'JPEG')

convert_pdf_to_jpeg('Freshres Plan.pdf', '/home/niranjan/Desktop/SecureKloud/
 ↪pdf to jpg/')
```

[24]:
```
import os
import img2pdf

def convert_images_to_pdf(image_folder, output_pdf_path):
    image_files = [filename for filename in os.listdir(image_folder) if␣
 ↪filename.endswith('.jpg')]
    image_paths = [os.path.join(image_folder, filename) for filename in␣
 ↪image_files]

    with open(output_pdf_path, "wb") as pdf_file:
        pdf_file.write(img2pdf.convert(image_paths))

convert_images_to_pdf('/home/niranjan/Desktop/SecureKloud/pdf to jpg/',
                      '/home/niranjan/Desktop/SecureKloud/pdf to jpg/output.
 ↪pdf')
```

0.0.7 Day - 7 and 8 Machine Learning Classification Metrics, ROC, AUC, etc..,

[1]:
```
print('ML model')
```

```
ML model
```

0.0.8 Day - 9 Python Encoding and Decoding

[27]:
```
original = 'Hello'

encoded = original.encode(encoding='utf-8')
print(type(encoded))

str_decoded = encoded.decode()
print(type(str_decoded))

print('Encoded bytes =', encoded)
print('Decoded String =', str_decoded)
print('str_original equals str_decoded =', original == str_decoded)
```

```
<class 'bytes'>
<class 'str'>
Encoded bytes = b'Hello'
Decoded String = Hello
str_original equals str_decoded = True
```

[31]:
```
def write_text_to_file(text, file_path, encoding='utf-8'):
    with open(file_path, 'w', encoding=encoding) as file:
        file.write(text)

def read_text_from_file(file_path, encoding='utf-8'):
    with open(file_path, 'r', encoding=encoding) as file:
        return file.read()

text = "Hi this is me"

write_text_to_file(text, 'utf8.txt', encoding='utf-8')
write_text_to_file(text, 'utf16.txt', encoding='utf-16')
write_text_to_file(text, 'utf32.txt', encoding='utf-32')
write_text_to_file('\ufeff' + text, 'utf8_bom.txt', encoding='utf-8-sig')

print("UTF-8:", read_text_from_file('utf8.txt', encoding='utf-8'))
print("UTF-16:", read_text_from_file('utf16.txt', encoding='utf-16'))
print("UTF-32:", read_text_from_file('utf32.txt', encoding='utf-32'))
print("UTF-8 with BOM:", read_text_from_file('utf8_bom.txt',␣
 ↪encoding='utf-8-sig'))
```

```
UTF-8: Hi this is me
```

```
UTF-16: Hi this is me
UTF-32: Hi this is me
UTF-8 with BOM: Hi this is me
```

[48]:
```
text = []
japanese_data = open('japanesetext.txt', 'r')
for i in japanese_data.readlines():
    text.append(i)

output = open('japanese_output_file.txt', 'w')
for i in text:
    output.writelines(i)

read_file = open('japanese_output_file.txt', 'r')
print(read_file.read())
```

```
:
:
:
:
:
:
:
:
:
:
:
:
:
:
:
:
```

0.0.9 Day - 10 Excel Operations, Functions, Data Analysis, Pivot table and Graphs

[49]:
```
# Excel basics - in excel app
```