10.1) Basics

Use "pwd" command to Know the present working directory

In [13]:

```
pwd
```

Out[13]:

```
'C:\\Users\\Euphor\\Documents\\Data_Science\\1_Python_Basics'
```

Use "ls" To list all the files in present working directory

In [14]:

```
ls
```

```
 Volume in drive C has no label.
 Volume Serial Number is 5A56-FF5E

 Directory of C:\Users\Euphor\Documents\Data_Science\1_Python_Basics

12-03-2023  08:56    <DIR>          .
23-02-2023  14:55    <DIR>          ..
11-03-2023  20:09    <DIR>          .ipynb_checkpoints
21-02-2023  10:50            10,659 1) Basics.ipynb
12-03-2023  08:56             4,257 10) File_Manipulation.ipynb
23-02-2023  13:40            19,260 2) Control_Flow.ipynb
23-02-2023  13:42            40,313 3) Operators.ipynb
23-02-2023  13:43            24,798 4) String Manupilation.ipynb
23-02-2023  13:44            22,562 5) List.ipynb
23-02-2023  14:23            28,876 6) Tuples, Sets and Dictionaries.ipynb
23-02-2023  13:38            32,109 7) Functions.ipynb
03-03-2023  11:20           403,836 8) OPPS.ipynb
06-03-2023  13:28            39,216 9) Opps_concepts.ipynb
06-03-2023  14:26                21 File_Extension.py
12-03-2023  08:54    <DIR>          Resources
12-03-2023  08:50                 0 text.txt
11-03-2023  19:57               605 Untitled.ipynb
              13 File(s)        626,512 bytes
               4 Dir(s)  69,521,715,200 bytes free
```

In [15]:

```
#or
import os
os.listdir()
```

Out[15]:

```
['.ipynb_checkpoints',
 '1) Basics.ipynb',
 '10) File_Manipulation.ipynb',
 '2) Control_Flow.ipynb',
 '3) Operators.ipynb',
 '4) String Manupilation.ipynb',
 '5) List.ipynb',
 '6) Tuples, Sets and Dictionaries.ipynb',
 '7) Functions.ipynb',
 '8) OPPS.ipynb',
 '9) Opps_concepts.ipynb',
 'File_Extension.py',
 'Resources',
 'text.txt',
 'Untitled.ipynb']
```

10.1.2) Reading the File

In [18]:

```
f = open('text.txt','rt')
#'r'- for reading
#'t'- for text file
```

In [19]:

```
#reading the file
f.read()
#file is empty
```

Out[19]:

```
'What is data science?\nData science is the study of data to extract meani
ngful insights for business. It is a multidisciplinary approach that combi
nes principles and practices from the fields of mathematics, statistics, a
rtificial intelligence, and computer engineering to analyze large amounts 
of data. This analysis helps data scientists to ask and answer questions l
ike what happened, why it happened, what will happen, and what can be done
with the results.\n\nWhy is data science important?\nData science is impor
tant because it combines tools, methods, and technology to generate meanin
g from data. Modern organizations are inundated with data; there is a prol
iferation of devices that can automatically collect and store information.
Online systems and payment portals capture more data in the fields of e-co
mmerce, medicine, finance, and every other aspect of human life. We have t
ext, audio, video, and image data available in vast quantities.  '
```

In [20]:

```
f.read()
#this time the read function comes out to be empty because the courser is on the end of t
#there is no text to read.
#to solve this we need to close the file first.
```

Out[20]:

```
''
```

In [21]:

```
f.close()
```

In [24]:

```
#another method to read
f =open("text.txt",'r')
for i in f:
    print(i)
f.close() #must remember to close the file
```

```
What is data science?

Data science is the study of data to extract meaningful insights for busin
ess. It is a multidisciplinary approach that combines principles and pract
ices from the fields of mathematics, statistics, artificial intelligence,
and computer engineering to analyze large amounts of data. This analysis h
elps data scientists to ask and answer questions like what happened, why i
t happened, what will happen, and what can be done with the results.



Why is data science important?

Data science is important because it combines tools, methods, and technolo
gy to generate meaning from data. Modern organizations are inundated with
data; there is a proliferation of devices that can automatically collect a
nd store information. Online systems and payment portals capture more data
in the fields of e-commerce, medicine, finance, and every other aspect of
human life. We have text, audio, video, and image data available in vast q
uantities.
```

readline()

In [40]:

```
f = open("text.txt",'r')
f.readline()  #Helps in reading line
```

Out[40]:

```
'What is data science?\n'
```

In [41]:

```
f.readline()
f.readline()
f.readline()
f.readline()
# By calling 'readline()' n no. of times the function read n no. of lines.
```

Out[41]:

```
'Data science is important because it combines tools, methods, and technol
ogy to generate meaning from data. Modern organizations are inundated with
data; there is a proliferation of devices that can automatically collect a
nd store information. Online systems and payment portals capture more data
in the fields of e-commerce, medicine, finance, and every other aspect of
human life. We have text, audio, video, and image data available in vast q
uantities.  '
```

In [ ]:

```

```

10.1.2) Writing/Creating Files

'a'- Will append the end of the line 'w'- Will overwrite the existing file

In [43]:

```
#copying file
import shutil
shutil.copy('text.txt','Resources/File_Handeling.txt')
f = open('Resources/File_Handeling.txt','r')
for i in f:
    print(i)
f.close()
```

```
What is data science?

Data science is the study of data to extract meaningful insights for busin
ess. It is a multidisciplinary approach that combines principles and pract
ices from the fields of mathematics, statistics, artificial intelligence,
and computer engineering to analyze large amounts of data. This analysis h
elps data scientists to ask and answer questions like what happened, why i
t happened, what will happen, and what can be done with the results.



Why is data science important?

Data science is important because it combines tools, methods, and technolo
gy to generate meaning from data. Modern organizations are inundated with
data; there is a proliferation of devices that can automatically collect a
nd store information. Online systems and payment portals capture more data
in the fields of e-commerce, medicine, finance, and every other aspect of
human life. We have text, audio, video, and image data available in vast q
uantities.
```

Writing Content

In [54]:

```
#openinf file in write mode
f = open("Resources/File_Handeling.txt",'w')
f.write("This file is opened on write mode due to which the content is overwritten")
f.close()
```

In [55]:

```
f = open("Resources/File_Handeling.txt",'r')
print(f.read())
f.close()
```

```
This file is opened on write mode due to which the content is overwritten
```

Appending content

In [56]:

```
#appending
f=open('Resources/File_Handeling.txt','a')
f.write("this is the content added at the end of the line because the file is onpened in
f.close()
```

In [57]:

```
#now reading
f = open("Resources/File_Handeling.txt",'r')
print(f.read())
f.close()
```

```
This file is opened on write mode due to which the content is overwrittent
his is the content added at the end of the line because the file is onpene
d in append mode('a')
```

Create a New File

Files can be created using following modes

> 'a'- Append - will create a file if the specified file does not exist

> 'w'- Write - will create a file if the specified file does not exist

> 'x'- Create - will create a file, returns an error if the file exist

In [58]:

```
#creating file using different modes
f1 = open("Resources/Crea5ted_using_x.txt",'x')
f2 = open("Resources/Crea5ted_using_a.txt",'a')
f3 = open("Resources/Crea5ted_using_w.txt",'w')
```

In [60]:

```
os.listdir('Resources')
#we can can see that the abouve 3 files have bee created
```

Out[60]:

```
['.ipynb_checkpoints',
 'Crea5ted_using_a.txt',
 'Crea5ted_using_w.txt',
 'Crea5ted_using_x.txt',
 'File_Handeling.txt']
```

10.1.3) Removing file

In [2]:

```
import os
os.listdir()
```

Out[2]:

```
['.ipynb_checkpoints',
 '1) Basics.ipynb',
 '10) File_Manipulation.ipynb',
 '2) Control_Flow.ipynb',
 '3) Operators.ipynb',
 '4) String Manupilation.ipynb',
 '5) List.ipynb',
 '6) Tuples, Sets and Dictionaries.ipynb',
 '7) Functions.ipynb',
 '8) OPPS.ipynb',
 '9) Opps_concepts.ipynb',
 'File_Extension.py',
 'Resources',
 'text.txt',
 'Untitled.ipynb']
```

In [3]:

```
#removing a file
os.remove("File_Extension.py")
```

In [4]:

```
#creating a folder/directory in python
os.mkdir('Resources/Folder_created_using_py')
```

In [ ]:

```

```

In [ ]:

```

```

10.2) Reading and Writing files

JSON file - Java-Script Object Notation

> Stores data as a key and value pair(similar to a dictionary)

In [17]:

```
#creating a json file
data = {
    'name':'asit',
    "mail_id" : "mail@mail.com",
    "phone_number":999995566,
    "subject":["Data science","DSA"]
}
```

In [18]:

```
import json
```

In [19]:

```
#"json.dump"- for write operation
with open("test.json","w") as f:
    json.dump(data,f)
```

In [20]:

```
#"json.load operation for write operation"
with open("test.json","r") as f:
    data1 =json.load(f)
```

In [21]:

```
data1
```

Out[21]:

```
{'name': 'asit',
 'mail_id': 'mail@mail.com',
 'phone_number': 999995566,
 'subject': ['Data science', 'DSA']}
```

In [23]:

```
#extracting DSA
data1["subject"][1]
```

Out[23]:

```
'DSA'
```

In [ ]:

```

```

CSV File- Comma Seperated Value File

In [24]:

```
dtat_csv = [
    ["name","emil","number"],
    ["asit","mail@mail.com","9988776655"],
    ["pra","pra@buura.com","5544112233"],
    ["din","din@shin.com","5566442288"]
]
```

In [25]:

```
import csv
```

In [26]:

```
with open ("test.csv","w") as f:
    w = csv.writer(f)
    for i in dtat_csv:
        w.writerow(i)
```

In [27]:

```
with open('test.csv',"r") as f:
    read = csv.reader(f)
    for i in read:
        print(i)
```

```
['name', 'emil', 'number']
[]
['asit', 'mail@mail.com', '9988776655']
[]
['pra', 'pra@buura.com', '5544112233']
[]
['din', 'din@shin.com', '5566442288']
[]
```

In [ ]:

```

```

Binary Data- ".bin" file extension

In [28]:

```
with open("test.bin","wb") as f:   #"wb"- write binary mode
    f.write(b"\x01\x02\x03")
```

In [30]:

```
with open("test.bin","rb") as f:
    print(f.read())
```

```
b'\x01\x02\x03'
```

In [ ]:

In [ ]:

10.3) Buffer read and write Operation

> Suppose we have to read a big file (gb or tb), we can't read it at a time we have read it in a small chuncks.

In [35]:

```
import io
```

In [39]:

```
#io.bufferwriter
with open("text_buffer.txt","wb") as f:
    file= io.BufferedWriter(f) #press shift+tab:- we can see that by diffault this burref
    file.write(b"this is my buffer write")
    file.write(b"this is my second line that i'm trying to write")
    file.flush() #closes file
#until we close the file there will be nothing written in it
```

In [43]:

```
with open("text_buffer.txt","rb") as f:
    file = io.BufferedReader(f)
    data= file.read(10) #reads only 10 bytes
    print(data)
```

```
b'this is my'
```

In [ ]:

In [ ]:

```

```

10.4) Logging and Debugging

Logging

> In a prduction grade code we are not suppose to use a print statement, as print sta5tement will always try to print something inside our console and once we shut-down our entire machine this will be gone.

> So we require a permanent storatge where we can log each and every information.

In [116]:

```
import logging
```

Logging Level Hierarchy

> 1- noset (no longer in python)

> 2- DEBUG

> 3- INFO

> 4- WARNING

> 5- ERROR

> 6- CRITICAL

if loggin level is set to warning then only warning and its lower levels wiull be logged in the file.

In [1]:

```
import logging
logging.basicConfig(filename="test.log", level= logging.INFO, format= "%(asctime)s %(name
```

In [2]:

```
logging.info("this is an info")
logging.debug("this is a debug") #this will not print as debug's hierarchy is higher than
logging.critical("this is critical")
logging.warning("this is warning")
```

In [ ]:

```

```

In [10]:

```
#Create a function to seperate the integers and string from the following list
#Create a log of every step of the program
lst = [11,3,5,7,[2,4,6],"asit",'shastri']
lst_int=[]
lst_str=[]
logging.info("---------------------------------------------------------------------------------------
for i in lst:
    logging.info("this is the start of my first for loop {}".format(lst))
    logging.info("this is the value i am logging {}".format(i))
    if type(i) == list:
        for j in i:
            logging.info("logging my j {j} and i is {i}".format(j=j,i=i))
            if type(j)==int:
                lst_int.append(j)
    elif type(i)==int:
        lst_int.append(i)
    else:
        lst_str.append(i)
logging.info("this is my final result with all int:-{} and with all string:-{}".format(ls

#So logging method help us know at what point the program has crashed
```

Always try to use logging in a production grade code

In [ ]:

```

```

In [ ]:

```

```