File handling

write mode

In [158]:
```
f=open("mydata","w")
```

In [159]:
```
f.write("hi my name is vaishnavi abbugari, my age is 22 "
        "and i have done my bachelors in microbilogy "
        "i want to do a transistion in data science "
        "as we all know its increasing graph ")
```

Out[159]:
```
170
```

In [160]:
```
f.close()
```

read mode

In [161]:
```
f=open("mydata","r")
```

In [162]:
```
f.read()
```

Out[162]:
```
'hi my name is vaishnavi abbugari, my age is 22 and i have done my bachelors in
microbilogy i want to do a transistion in data science as we all know its incre
asing graph '
```

In [163]:
```
#here we cant see any output because the pointer came to end
print(f.read())
```

In [164]:
```
#so to change the position of the pointer at 1st we have to use seek functio
f.seek(0)
```

Out[164]:
```
0
```

In [165]:
```
f.read()
```

Out[165]:
```
'hi my name is vaishnavi abbugari, my age is 22 and i have done my bachelors in
microbilogy i want to do a transistion in data science as we all know its incre
asing graph '
```

apend mode

In [175]:
```
f=open("mydata","a")
```

```
In [176]: f.write("currently iam learning fullstack data science course")

Out[176]: 52
```

```
In [177]: f.close()
```

```
In [178]: f=open("mydata","r")
```

```
In [179]: f.read()

Out[179]: 'hi my name is vaishnavi abbugari, my age is 22 and i have done my bachelors in
          microbilogy i want to do a transistion in data science as we all know its incre
          asing graph currently iam learning fullstack data science coursethis will delet
          e all the previous dta and it will overridecurrently iam learning fullstack dat
          a science course'
```

a+(appends,writes and reads)

```
In [198]: f=open("mydata","a+")
```

```
In [199]: f.seek(0)

Out[199]: 0
```

```
In [200]: f.read()

Out[200]: 'hajg'
```

```
In [201]: f.write("ha")

Out[201]: 2
```

```
In [202]: f.close()
```

w+(writes and reads)

```
In [203]: f=open("mydata","w+")
```

```
In [207]: f.seek(0)

Out[207]: 0
```

```
In [208]: f.read()

Out[208]: 'jg'
```

In [209]:

```
f.write("jg")
```

Out[209]: 2

r+(reads and writes)

In [210]:

```
f=open("mydata","r+")
```

In [211]:

```
f.read()
```

Out[211]: 'jgjg'

In [212]:

```
f.write("thank you")
```

Out[212]: 9

In [213]:

```
f.close()
```

Threading

In [223]:

```
from threading import *
```

In [227]:

```
#threading
```

In [226]:

```
class demo:
    def show():
        for i in range(5):
            print("this is the child thread")
obj=demo
t=Thread(target=obj.show())
t.start()
for i in range(5):
    print("this is the parent thread")
```

```
this is the child thread
this is the child thread
this is the child thread
this is the child thread
this is the child thread
this is the parent thread
this is the parent thread
this is the parent thread
this is the parent thread
this is the parent thread
```

In [228]:

```
#multithreading
```

In [235]:

```
from threading import *
class Demo():
    def num(self):
        for i in range(6):
            print("the number is :",i)

    def doubled(self):
        for i in range(6):
            print("the doubled number is :",2*i)

    def sqrt(self):
        for i in range(6):
            print("the sqrt number is :",i*i)

obj=Demo()
t1=Thread(target=obj.num)
t2=Thread(target=obj.doubled)
t3=Thread(target=obj.sqrt)
t1.start()
t2.start()
t3.start()

```

```
the number is : 0
the number is : 1
the number is : 2
the number is : 3
the number is : 4
the number is : 5
the doubled number is : 0
the doubled number is : 2
the doubled number is : 4
the doubled number is : 6
the doubled number is : 8
the doubled number is : 10
the sqrt number is : 0
the sqrt number is : 1
the sqrt number is : 4
the sqrt number is : 9
the sqrt number is : 16
the sqrt number is : 25
```

Exception handling

In [236]:

```
a=10
```

In [238]:

```python
a=b
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [238], in <cell line: 1>()
----> 1 a=b

NameError: name 'b' is not defined
```

In [241]:

```python
#to overcome the above errors we will be using exception handling
try:
    a=b
except Exception as e:
    print(e)
```

```
name 'b' is not defined
```

In [248]:

```python
try:
    a=int(input("enter your first number :"))
    b=int(input("enter your second number :"))
except Exception as e:
    print(e)
else:
    print(a+b)
    print(a-b)
    print(a*b)
    print(a/b)
```

```
enter your first number :197489
enter your second number :7254
204743
190235
1432585206
27.22484146677695
```

In [249]:

```python
try:
    a=int(input("enter your first number :"))
    b=int(input("enter your second number :"))
except Exception as e:
    print(e)
else:
    print(a+b)
    print(a-b)
    print(a*b)
    print(a/b)
finally:
    print('execution is done')
```

```
enter your first number :10
enter your second number :20
30
-10
200
0.5
execution is done
```

In [256]:

```python
try:
    a=int(input("enter your first number :"))
    b=int(input("enter your second number :"))

except Exception as e:
    print(e)
else:
    print(a+b)
    print(a-b)
    print(a*b)
finally:
    print('execution is done')
```

```
enter your first number :10
enter your second number :0
10
10
0
execution is done
```

Mail automation

In [268]:

```python
#simple mail transfer protocol
import smtplib
smtobj=smtplib.SMTP("smtp.gmail.com",587)
smtobj.ehlo()
smtobj.starttls()
smtobj.login("your mail id","password")
smtobj.sendmail("your mail id","persons mail id","message")
smtobj.quit()
```