Opening a file:

To open a file we simply use open command and specify the 'path' and the 'mode'. With the direct open command we have to close the file otherwise an instance of the file remains open.

In [1]:
```
f = open(r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\FileR
print (f.name)
f.close()
```

```
C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\FileRw.txt
```

Opening a file with context manager

As seen above I have used the f.close() method to close the file. In order to close the file automatically once the operation is done (or any exception) we use with command as below

In [34]:
```
with open (r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\Fil
    pass
print (f.closed) #to check whether the file is closed, it returns True
```

```
True
```

Reading a file:

f.read() reads the file completely,

In [1]:
```
with open (r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\Fil
    read_ = f.read()
#     readlines_ = f.readlines()
    print (read_)
print (f.closed)
```

```
1) This is first line
2) This is second line
3) This is third line
4) This is fourth line
5) This is fifth line
6) This is sixth line
7) This is seventh line
True
```

To read only a specific characters passed as an argument. If we go on excuting the same command multiple times, it goes on printinf next and next line.

In [23]:
```
with open (r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\Fil
    read_ = f.read(44)
    print (read_,end='*')
    read_ = f.read(44)
    print (read_,end='*')
    read_ = f.read(44)
    print (read_,end='*')
#     read_ = f.read(44)
#     print (read_,end='')
f.close()
```

```
1) This is first line
2) This is second line*
3) This is third line
4) This is fourth lin*e
5) This is fifth line
6) This is sixth lin*
```

In [25]:
```
with open (r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\Fil
    readlines_ = f.readlines()
print(readlines_) #provides a list of all lines
f.close()
```

```
['1) This is first line\n', '2) This is second line\n', '3) This is third line
\n', '4) This is fourth line\n', '5) This is fifth line\n', '6) This is sixth l
ine\n', '7) This is seventh line']
```

But in case if we want to read the file partially only and not the whole file then we have other methods like readline()

In [26]:
```
with open (r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\Fil
    readline_ = f.readline()
    print(readline_,end='') #provides a single line
    readline_ = f.readline()
    print(readline_,end='')
    readline_ = f.readline()
    print(readline_,end='')
    #I can print multiple lines by calling the print function repeatedly
f.close()
```

```
1) This is first line
2) This is second line
3) This is third line
```

Using For and while loops to read a file:

In [4]:
```
with open (r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\Fil
    for i in f:
        print (i,end='')
```

```
1) This is first line
2) This is second line
3) This is third line
4) This is fourth line
5) This is fifth line
6) This is sixth line
7) This is seventh line
```

In [2]:
```
#Reading a file using a while loop:
```

In [28]:
```
with open (r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\Fil
    f_contents = f.read(22)
    while len(f_contents)>0:
        print (f_contents,end='*')
        f_contents = f.read(22)
```

```
1) This is first line
*2) This is second line*
3) This is third line*
4) This is fourth lin*e
5) This is fifth lin*e
6) This is sixth lin*e
7) This is seventh l*ine*
```

f.seek() and f.tell

f.tell() : it tells us the current location of the cursor in a file while reading a file

f.seek() : brings the cursor to the position specified in the argument and starts printing from there again

In [29]:
```
with open (r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\Fil
    f_contents = f.read(22)
    print (f_contents)
    print (f.tell()) #after reading the 22 characters, the cursor is located
#     f.seek(12)
#     f_contents = f.read(22)
#     print (f_contents)
```

```
1) This is first line

23
```

Writing a file:

For reading a file we need to have a file at desired location, but for writing a file even if the file is not there the write will create a file and start inserting text in it.

In [39]:
```
with open (r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\Fil
    f.write('This is a demo write function\n')
    f.write('This is a demo Read write function')
    #if I try to run the write function multiple times it will go on writing
```

Copying contents of one file in the other file:

In [34]:
```
with open(r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\File
    with open(r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\
        for lines_ in rf:
            wf.write(lines_)


```

In [33]:
```
#1)
# with open(r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\wa
#     with open(r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Datase
#         for lines_ in rf:
#             wf.write(lines_)
#2)
with open(r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\watc
    with open(r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\
        size = 30
        content = rf.read(size)
        while len(content)>0:
            wf.write(content)
            content = rf.read(size)

```

Deleting a file:

In [31]:
```
#For delelting a file we need to import os(operating system) module as we ar
import os
os.remove(r"C:\Users\admin\Desktop\Python Training\Numpy Pandas Dataset\File
```

Practise Questions:

In [ ]:
```
# Create one text file 'practise' enter some text and read it by using read(
# copy the contents of this file to other file 'practise_copy'
# using the same file operation copy any random image
```