File handling in python

> Today we will learn the concept of file in python
>
> 1. How to open a file
> 2. How to read a file
> 3. How to write or update a file and many more things in python

Syntax:

```
    open('filename','mode')  --> return a file object point to a file
```

But first let us go through the **file modes**
'r' - Read - Opens a file only for reading, default mode, returns error if file not exists
'w' - Write - File only for writing, creates a file if not exists, delete all old data.
'a' - Append - Writing in a file, creates a file if not exists, append data at the end of the previous data if any **'w+** - Both for read and write

Additional modes

't' - text mode **'b'** - binary files (like images)

In [1]:
```
file = open('filename1.txt', 'r')
## this should return error, since file does not exists
```

```
---------------------------------------------------------------------
----
FileNotFoundError                         Traceback (most recent call l
ast)
<ipython-input-1-b4d44712d535> in <module>
```

```
----> 1 file = open('filename1.txt', 'r')
      2 ## this should return error, since file does not exists

FileNotFoundError: [Errno 2] No such file or directory: 'filename1.txt'
```

close a file

we should always close a file in order to use memory efficiently

f_obj.close()

In [2]:
```
file.close()
```

```
---------------------------------------------------------------------
----
NameError                                 Traceback (most recent call l
ast)
<ipython-input-2-2f72176549a0> in <module>
----> 1 file.close()

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

In [3]:
```
file = open('filename1.txt','w')
file.close()
```

In [4]:
```
file = open('filename1.txt','r')
file.close()
```

wow! no error

Some function for file handling

1. Read

read() - reads all the data from the file
read(characters) - read specified numbers of characters readline() - return one line at a time readlines() - return a list of all the lines in the file By looping through the file object

In [5]:
```
f = open('filename.txt','r')
print(f.read())
f.close()
```

```
hello hey every one
How are you?
Expecto patronum
Avada Kedavra
```

In [ ]:
```

```

In [6]:
```
f = open('filename.txt','r')
print(f.readline())
print(f.readline())
f.close()
```

```
hello hey every one

How are you?
```

Note : In the output it print a extra empty line Reason: Since there is a new line character at the ending of each line in the file and print() also adds \n at the end by default

In [7]:
```
f = open('filename.txt','r')
print(f.readlines())    # returns a list of lines
f.close()
```

```
['hello hey every one\n', 'How are you?\n', 'Expecto patronum\n', 'Avad
a Kedavra']
```

In [ ]:
```
# iterate over file object
```

In [8]:
```
f = open('filename.txt','r')
for line in f:
    print(line, end ='')
f.close()
```

```
hello hey every one
How are you?
Expecto patronum
Avada Kedavra
```

2. Write

> write('message') - write the passed message in the file and return the length of the message

In [9]:
```
f = open('filename2.txt','w')
f.write('Welcome to the Wonderland\n --By Anson Seabra')
f.close()
```

In [10]:
```
f = open('filename2.txt','r')
print(f.read())
f.close()
```

```
Welcome to the Wonderland
 --By Anson Seabra
```

3. Append

In [11]:
```
f = open('filename2.txt','a')
f.write('\nReleased: 2018\nGenre: Pop')
```

```
f.close()
```

In [12]:
```
f = open('filename2.txt','r')
print(f.read())
f.close()
```

```
Welcome to the Wonderland
 --By Anson Seabra
Released: 2018
Genre: Pop
```

There is one more ways to open a file

with open() as obj:

In this method you need not to close the file explicitly.

In [15]:
```
with open('filename2.txt') as f:
    print(f.read())
f.read()
```

```
Welcome to the Wonderland
 --By Anson Seabra
Released: 2018
Genre: Pop

---------------------------------------------------------------------
----
ValueError                                Traceback (most recent call l
ast)
<ipython-input-15-b3184076f938> in <module>
      1 with open('filename2.txt') as f:
      2     print(f.read())
----> 3 f.read()

ValueError: I/O operation on closed file.
```

In [14]:
```
f.read()
```

```
---------------------------------------------------------------------
---
ValueError                                Traceback (most recent call l
ast)
<ipython-input-14-571e9fb02258> in <module>
----> 1 f.read()

ValueError: I/O operation on closed file.
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: