Before we finish this section, lets quickly go over how to perform simple I/O with basic .txt files

In Jupiter, we can quickly write a text file like so:

In [1]:
```
%%writefile myfile.txt
Hello this is a text file
this is the second line
this is the third line
```

```
Writing myfile.txt
```

In [2]:
```
myfile = open('myfile.txt')
```

The file has to be saved/located in the same directory as the Jupiter notebook. We can confirm the location of our Jupiter notebook like so:

In [3]:
```
pwd
```

Out[3]:
```
'C:\\Users\\Keegz\\My Python Stuff'
```

In [4]:
```
myfile = open ('myfile.txt')
```

In [5]:
```
myfile.read()
```

Out[5]:
```
'Hello this is a text file\nthis is the second line\nthis is the third line\n'
```

The .read() reads the whole string within the text file, notice the \n, this indicates a new line, but since we asked for a single string, it provided just one single string

In [6]:
```
myfile.read()
```

Out[6]:
```
''
```

When you .read() a file, the cursor goes all the way to the end. We need to reset the cursor to the beginning of the file. We can do this like so:

In [7]:
```
myfile.seek(0)
```

Out[7]:
```
0
```

In [8]:
```
myfile.read()
```

Out[8]:
```
'Hello this is a text file\nthis is the second line\nthis is the third line\n'
```

The method method produces one giant string, which may not always be useful. Lets look at the read lines method

In [11]:
```
myfile.readlines()
```

Out[11]:
```
['Hello this is a text file\n',
 'this is the second line\n',
                      ird line\n']
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Lets discuss file locations.

If we want to open files at another location on your computer, simply pass in the entire file path. For Windows you need to use a double \ so Python doesnt treat the second \ as an escape character, a file path is in the form:

myfile = open("C:\Users\UserName\Folder\test.txt")

For MacOS and Linux you use slashes in the opposite direction:

myfile = open(" /Users/YouUserName/Folder/mylife.txt")

In [12]:
```
pwd
```
Out[12]:
```
'C:\\Users\\Keegz\\My Python Stuff'
```

Once we open a file, we also need to close it once completed, we can do this like so:

In [13]:
```
myfile.close()
```

In [14]:
```
with open ('myfile.txt') as my_new_file:
    contents = my_new_file.read()
```

We no longer have to worry about closing the file we the above command

In [15]:
```
contents
```
Out[15]:
```
'Hello this is a text file\nthis is the second line\nthis is the third line\n'
```

We can also write/overwrite files like so:

In [16]:
```
with open('myfile.txt',mode='r') as myfile:
    contents = myfile.read()
```

In [17]:
```
contents
```
Out[17]:
```
'Hello this is a text file\nthis is the second line\nthis is the third line\n'
```

Loading [MathJax]/extensions/Safe.js

Above, we see the mode was set to read, and it provided the contents. Now lets switch the mode to write (w) and see what happens:

In [18]:
```
with open('myfile.txt',mode='w') as myfile:
    contents = myfile.read()
```

```
---------------------------------------------------------------------------
UnsupportedOperation                      Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_7684/3330133697.py in <module>
      1 with open('myfile.txt',mode='w') as myfile:
----> 2     contents = myfile.read()

UnsupportedOperation: not readable
```

The error pictured above is due to permissions. Sometimes, we want the file to have both read and write permissions.

Reading, Writing, Appending Modes

mode='r' is read only mode= 'w' is write only (will overwrite files or create new!) mode='a' is append only (will add on to files) mode = 'r+' is reading and writing mode= 'w+' is writing and reading (Overwrites existing files or creates a new file!)

In [28]:
```
%%writefile my_new_file.txt
ONE ON FIRST
TWO ON SECOND
THREE ON THIRD
```

```
Overwriting my_new_file.txt
```

In [23]:
```
with open('my_new_file.txt' ,mode='r') as f:
    print(f.read())
```

```
ONE ON FIRST
TWO ON SECOND
THREE ON THIRD
```

Lets say we wanted to add a new line to our file, we can do so by performing the following:

In [29]:
```
with open('my_new_file.txt' ,mode='a')as f:
    f.write('FOUR ON FOURTH')
```

In [30]:
```
with open('my_new_file.txt' ,mode='r') as f:
    print(f.read())
```

```
ONE ON FIRST
TWO ON SECOND
THREE ON THIRD
FOUR ON FOURTH
```

Now, lets explore w (writing):

In [31]:
```
with open('asdfgghhhj.txt' ,mode='w') as f:
    f.write('I CREATED THIS FILE!')
```

Since we opened a non-existant file with 'w', it created the file instead of throwing an error. Any

Loading [MathJax]/extensions/Safe.js

other mode, r, a would have thrown an error

In [32]:
```
with open ('asdfgghhhj.txt',mode='r') as f:
     print(f.read())
```

```
I CREATED THIS FILE!
```

File I/O Practice

Write a script that opens a file named 'test.txt', writes 'Hello World' to the file, then closes it.

x = open('test.txt', 'w') x.write('Hello World') x.close()

In [ ]:
```

```