

SECTION 3: PYTHON OBJECT AND DATA STRUCTURE BASICS, 2 hrs 2 mins, 33 parts

- **32/36 I/O with Basic Files in Python**
 - **-> I/O for text files (loading them into a JN)**
 - **in the JN**
 - **-> create a text file to work with**
 - **%%writefile myfile.txt**
 - **Insert the text for the file here**
 - **-> run the cell and it saves the txt file**
 - **-> myfile = open('myfile.txt')**
 - **if there is no file there, then it returns Errno 2**
 - **-> either the file doesn't exist, or the file isn't saved into the same directory as the JN**
 - **-> pwd in a cell returns the directory of the JN**
 - **the txt file needs to be saved into the same directory as the JN**
 - **-> then myfile = open('myfile.txt')**
 - **myfile.read() <- this returns a string of everything in the txt file, \n indicates new lines (in the txt file returned as an str) -> these are escape sequences**
 - **-> so, you open the txt file**
 - **-> store it as a variable**
 - **-> then run the read method on that variable**
 - **-> and the txt file had to be in the same working directory as the JN**
 - **-> myfile.read() again returns an empty string**
 - **-> there is a cursor in the file -> and when you read it it does to the end**
 - **-> set the cursor to the beginning of the file**
 - **myfile.seek(0)**
 - **-> then if you run read again -> it's outputting the contents of the entire file**
 - **-> to return the contents of the file in a list (line by line)**
 - **myfile.seek(0) <- return the cursor to the beginning**
 - **myfile.readlines() <- this returns the contents of the file line by line (each line an element in a string)**
 - **-> file locations**
 - **to open a txt file saved at another location on the computer**
 - **-> myfile = open("path_to_file")**
 - **-> best practices for opening files**
 - **-> myfile is the name of the file -> you need to close the file to not get errors (e.g if you are deleting the file later and it's still open elsewhere)**
 - **-> myfile.close()**
 - **-> with open('myfile.txt') as my_new_file: <- you are opening it as a new variable name**
 - **contents = my_new_file.read() <- then you don't need to worry about closing the file**
 - **-> reading and writing to a file**
 - **with open('myfile.txt',mode='r') as myfile:**
 - **contents = myfile.read()**
 - **-> he clicks shift and tab, and then documentation shows in the JN**
 - **there is a default mode called r**
 - **-> if you choose w as the mode -> permissions**
 - **-> depending on the permissions, you can get error messages**
 - **-> to overwrite files / append to them etc**
 - **-> reading / writing / appending modes**
 - **the different values of mode**

- **r <- read only**
- **w <- write only**
- **a <- append only (adds onto files)**
- **r+ <- reading and writing (this overwrites the file if it exists)**
- **w+ <- writing and reading**
- **thought process**
 - he writes a txt file
 - then opens it -> with `open('my_new_file.txt',mode='r')` as f:
 - `print(f.read())`
 - append mode
 - **with `open('my_new_file.txt',mode='a')` as f:**
 - **`f.write('\n FOUR ON FOURTH')` <- the cursor is at the end, it's adding this onto the end of the file (because its been set into append mode)**
 - write mode
 - **with `open('my_new_file.txt',mode='w')` as f:**
 - **`f.write('I CREATED THIS FILE!')` <- write mode, it's overwriting an existing file or creating it if it doesn't exist (in a different mode this would have returned an error)**
 - read mode
 - **with `open('my_new_file.txt',mode='r')` as f:**
 - **`print(f.read())` <- it prints the content of the file**