[Day 7]

File operation in Python

File handling is a crucial part of any programming language. Python provides built in functions & methods to read from & write to files both text & binary

Read file line by line

```
with open('example.txt','r') as file:
    for line in file:
        print(line.strip())  ## strip() removes the newline character
```

Writing file

```
with open('example.text','w') as file:
    file.write("Hello world! \n")
    file.write('this is new line.')
```

Writing file (without overwriting)

```
with open('example.txt','a') as file
    file.write('Append operation!')
```

Writing a list of lines to a file

```
lines = ['first line \n', 'second line \n', 'Third line \n']
with open('example.txt','x') as file:
    file.writelines(lines)
```

For Binary files: wb, rb

[Note:] The 'w+' mode in Python is used to open a file for both reading and writing. If the file does not exist, it will be created. If the file exists, its content is truncated (i.e. file is overwritten)

Writing & then reading a file

```
with open('example.txt', 'w+') as file:
    file.write("Hello world \n")
    file.write("This is a new line \n")

    # Move cursor to the beginning
    file.seek(0)

    # Read the content of the file
    content = file.read()
    print(content)
```