

# Python Files operations

## With open:

- ❖ used to open a file in a way that ensures the file is properly closed after its suite finishes, even if an error occurs during the execution

```
1 with open('filename.txt', 'mode') as file:
2     # Code to work with the file goes here
3     # ...
```

### File Modes for Text Files:

#### 'r' (read):

- Opens an existing file for reading.
- Raises an error if the file doesn't exist.
- Default mode if none is specified.

#### 'w' (write):

- Opens a file for writing.
- Creates a new file if it doesn't exist, or overwrites an existing file.

#### 'a' (append):

- Opens a file for appending (adding content to the end).
- Creates a new file if it doesn't exist.

#### 'x' (exclusive creation):

- Creates a new file for writing.
- Raises an error if the file already exists.

#### 'r+' (read and write):

- Opens an existing file for both reading and writing.
- Raises an error if the file doesn't exist.

#### 'w+' (write and read):

- Opens a file for both writing and reading.
- Creates a new file if it doesn't exist, or overwrites an existing file.

#### 'a+' (append and read):

- Opens a file for both appending and reading.
- Creates a new file if it doesn't exist.

### File Modes for Binary Files:

#### 'rb' (read binary):

- Opens an existing binary file for reading.
- Raises an error if the file doesn't exist.

#### 'wb' (write binary):

- Opens a binary file for writing.
- Creates a new file if it doesn't exist, or overwrites an existing file.

#### 'ab' (append binary):

- Opens a binary file for appending.
- Creates a new file if it doesn't exist.

#### 'rb+' (read and write binary):

- Opens an existing binary file for both reading and writing.
- Raises an error if the file doesn't exist.

#### 'wb+' (write and read binary):

- Opens a binary file for both writing and reading.
- Creates a new file if it doesn't exist, or overwrites an existing file.

#### 'ab+' (append and read binary):

- Opens a binary file for both appending and reading.
- Creates a new file if it doesn't exist.

## readlines() method:

❖ This method reads all the lines from the file and returns them as a list of strings.

```
1 with open('logfile.txt', 'r') as file:
2     lines = file.readlines()
3     for line in lines:
4         print(line)
```

## Use Cases:

### Search for a word in lines then Replace this line in txt file

- We are using this to overwrite a line in the file
- But we can use it to edit the line too

```
1 with open('logfile.txt', 'r') as file:
2     lines = file.readlines()
3
4 for index, line in enumerate(lines):
5     if "name" in line:
6         print("Line index", index)
7         print("Line Content=", line)
8         lines[index] = "my name is Mohamed Hamed\n" # Modify the line in the list
9
10 with open('logfile.txt', 'w') as file:
11     file.writelines(lines)
```

### Change line in binary file :

```
1 with open('file.bin', 'rb+') as file:
2     lines = file.readlines()
3
4 for index, line in enumerate(lines):
5     if index == 2: # go to Specfic line
6         line_inbytes = bytearray(line) # Convert bytes to bytearray so we can modify it
7
8         line_inbytes[1:10] = b'66'      # Change only the first 10 digits of the bytes
9         line_inbytes = b'66'          # or we can overwrite the content of line to b'66'
10
11         lines[index] = line_inbytes    # apply the changes to the specific line in all lines
12         break
13
14
15 with open('file.bin', 'wb') as file:
16     file.writelines(lines)            # write the lines back with the changed line
```

## Remove Empty Lines:

```
1 with open('logfile.txt', 'r') as file:      # Read from the File
2     lines = file.readlines()
3
4 # Looping through the list `lines` in reverse order to efficiently remove empty lines
5 for index in range(len(lines) - 1, -1, -1): # Loop in the list backward: range(Start=size-1, Stop=-1, Step=-1)
6     if lines[index].strip() == "":          # Check if the line at index is empty after stripping whitespace
7         del lines[index]                   # If empty, delete the line from the list
8
9 with open('logfile.txt', 'w') as file:      # Overwrite the file with the new changes
10     file.writelines(lines)
```

## Insert new Line Before a specific line:

We can insert the line after it by incrementing the index by 1 (index+1)

```
1 with open('logfile.txt', 'r') as file:      # Read from the File
2     lines = file.readlines()
3
4 # Iterate over a copy of the list `lines` using enumerate to access both index and line
5 line_to_insert = "insert This Line\n"
6 Required_word = "Mohamed"
7
8 for index, line in enumerate(lines[:]):
9     if Required_word in line:
10         print(f"found Required Line with Index = {index}")
11         lines.insert(index, line_to_insert)
12
13 with open('logfile.txt', 'w') as file:      # Overwrite the file with the new changes
14     file.writelines(lines)
```

We can use the same methods as above example to search for a line and delete it:

```
1 del lines[Req_index]
```