File Handling in Python: Text Files

1. Opening a Text File

To work with a text file in Python, we use the open() function.

Syntax:

file_object = open("filename.txt", mode)

- file_object: Variable that holds the file reference.

- filename.txt: Name of the file (including path if needed).

- mode: Specifies the purpose (read, write, append, etc.).

3. Closing a Text File

After performing file operations, always close the file using close().

Syntax:

file_object.close()

Example:

```
f = open("demo.txt", "r")
# Perform operations
f.close()
```

Why Close Files?

- Releases system resources.

- Ensures all data is properly written.

2. Text File Open Modes

Mode	Description
`r`	**Read** – Opens file for reading (default mode). Raises error if file does not exist.
`r+`	**Read & Write** – Opens file for both reading and writing. Raises error if file does not exist.
`w`	**Write** – Opens file for writing. Creates a new file if it doesn't exist. **Overwrites** existing content.
`w+`	**Write & Read** – Opens file for both reading and writing. Creates a new file if it doesn't exist. **Overwrites** existing content.
`a`	**Append** – Opens file for appending data. Creates a new file if it doesn't exist. Preserves existing content.
`a+`	**Append & Read** – Opens file for both appending and reading. Creates a new file if it doesn't exist. Preserves existing content.

4. Opening a File Using with Clause

The with statement automatically closes the file after the block ends.

Syntax:

```
with open("filename.txt", mode) as file_object:
    # File operations
```

Example:

```
with open("demo.txt", "r") as f:
data = f.read()
print(data)
# File is automatically closed here
```

5. Writing/Appending Data to a Text File

(a) **write() Method:** Writes a single string to the file.

Example:

```
with open("demo.txt", "w") as f:
f.write("Hello, World!")
```

(b) writelines() Method: Writes a list of strings to the file.

Example:

```
lines = ["First line\n", "Second line\n"]
with open("demo.txt", "w") as f:
f.writelines(lines)
```

6. Reading from a Text File

(a) read() Method: Reads the entire file content as a single string.

Example:

```
with open("demo.txt", "r") as f:
content = f.read()
print(content)
```

(b) readline() Method: Reads one line at a time.

Example:

```
with open("demo.txt", "r") as f:
    line1 = f.readline()
    line2 = f.readline()
print(line1, line2)
```

(c) readlines() Method: Reads all lines and returns them as a list.

Example:

```
with open("demo.txt", "r") as f:
lines = f.readlines()
for line in lines:
print(line)
```

7. seek() and tell() Methods

(a) tell() : Returns the current position of the file pointer within a file.

Example:

```
with open("demo.txt", "r") as f:
print(f.tell())  # Output: 0 (start of file)
f.read(5)
print(f.tell())  # Output: 5 (after reading 5 chars)
```

(b) seek(offset, whence) : Moves the file pointer to a specified position.

- **offset**: Number of bytes to move.
- **whence**: Reference point (0=start, 1=current, 2=end).

Example:

```
with open("demo.txt", "r") as f:
f.seek(5)  # Move to 5th byte
print(f.read())
```

8. Manipulation of Data in a Text File:

(i) Updating a Record in a file:

```
with open("students.txt", "r+") as f:
data = f.readlines()
f.seek(0)
```

```
for line in data:
if "John" in line:
line = line.replace("John", "Jonathan")
f.write(line)
```

(ii) Example: Deleting a Record from file:

```
with open("students.txt", "r") as f:
lines = f.readlines()
with open("students.txt", "w") as f:
for line in lines:
if "Mark" not in line:
f.write(line)
```