

Python File Input & Output (File I/O)

Detailed Study Notes

1 Introduction to File I/O

File Input / Output ka matlab hai file se data read karna (input) aur file me data write karna (output). Python me files ka use tab hota hai jab hume data permanently store karna ho ya program band hone ke baad bhi data chahiye ho.

Real life example: Notebook ko open karna, usme likhna ya parhna, aur phir band kar dena.

2 open() Function

Python me file handle karne ke liye `open()` function use hota hai.

Syntax:

```
file = open("filename.txt", "mode")
```

Parameters:

- filename: File ka naam
- mode: File kis kaam ke liye open hogi

3 File Modes

- r – Read mode
- w – Write mode (overwrite)
- a – Append mode
- x – New file create karna
- rb – Read binary
- wb – Write binary

4 Reading Data from File (Read Mode)

Read mode me file ka data parhna possible hota hai. Agar file exist nahi karti to error aata hai.

Example 1: `read()`

```
file = open("data.txt", "r")
print(file.read())
file.close()
```

Is example me puri file ek sath read ho jati hai.

Example 2: `read limited characters`

```
file = open("data.txt", "r")
print(file.read(5))
file.close()
```

Sirf 5 characters read honge.

Example 3: `readline()`

```
file = open("data.txt", "r")
print(file.readline())
file.close()
```

Sirf pehli line read hoti hai.

Example 4: `readlines()`

```
file = open("data.txt", "r")
print(file.readlines())
file.close()
```

Output list ke form me hota hai.

Example 5: `for loop se read`

```
file = open("data.txt", "r")
for line in file:
    print(line)
file.close()
```

Yeh large files ke liye best method hai.

5 Writing Data to File (Write Mode)

Write mode me file ka purana data delete ho jata hai aur naya data likha jata hai.

Example 1: write()

```
file = open("output.txt", "w")
file.write("Hello Python")
file.close()
```

Example 2: multiple lines

```
file = open("output.txt", "w")
file.write("Line 1\n")
file.write("Line 2\n")
file.close()
```

Example 3: user input write

```
name = input("Enter name: ")
file = open("names.txt", "w")
file.write(name)
file.close()
```

Example 4: loop ke sath write

```
file = open("numbers.txt", "w")
for i in range(5):
    file.write(str(i) + "\n")
file.close()
```

6 Append Mode

Append mode me data file ke end me add hota hai aur purana data delete nahi hota.

Example 1

```
file = open("log.txt", "a")
file.write("\nNew Entry")
file.close()
```

Example 2

```
with open("data.txt", "a") as f:
    f.write("Append Mode\n")
```

7 with Statement

with statement ka faida yeh hai ke file automatically close ho jati hai.

Example 1

```
with open("data.txt", "r") as file:
    print(file.read())
```

Example 2

```
with open("output.txt", "w") as file:
    file.write("Auto Close Example")
```

8 Common Errors

- FileNotFoundError – File exist nahi karti
- PermissionError – Write permission nahi hoti
- ValueError – Galat mode use kiya jata hai

9 Conclusion

Python File I/O se hum data ko permanently store aur retrieve kar sakte hain. `with` statement best practice hai aur append mode data loss se bachata hai.