FILE HANDLING IN PYTHON

📘 1. What is File Handling?

Definition:
File handling in Python refers to the process of **creating, reading, writing, modifying, and deleting files** stored on a computer's disk using Python's built-in functions.

Purpose:
When a program ends, all its data stored in memory (RAM) is lost.
To make data *permanent*, we store it in a **file** on the hard drive.

In simple terms:

File handling = Saving and managing data permanently outside your program.

⚙️ 2. Opening Files with open() Function

🧩 Syntax:

file_object = open(filename, mode, buffering, encoding)

Parameters:

Parameter	Description
filename	The name (and path) of the file to open.
mode	The mode in which to open the file — determines if it's for reading, writing, etc.
buffering	(Optional) Controls how data is buffered in memory before writing.
encoding	Used for encoding text (e.g., utf-8). Default is platform dependent.

✅ Returns:

A **file object**, which can be used to read, write, or manipulate the file.

📜 3. File Modes — Complete Table with Examples

Let's understand all **possible file opening modes** in Python — including beyond the basics.

Mode	Full Form	Description	Pointer Position	File Must Exist?	Example
'r'	Read	Opens file for reading (default mode).	Start of file	✅ Yes	open("data.txt", "r")
'w'	Write	Opens file for writing. Overwrites existing file or creates new one.	Start of file	❌ No	open("data.txt", "w")
'a'	Append	Opens file for appending data to the end. Keeps old content.	End of file	❌ No	open("data.txt", "a")
'r+'	Read + Write	Opens file for both reading and writing. Does **not truncate** file.	Start of file	✅ Yes	open("data.txt", "r+")
'w+'	Write + Read	Opens file for both writing and reading. **Truncates** file if it exists.	Start of file	❌ No	open("data.txt", "w+")
'a+'	Append + Read	Opens file for both appending and reading. Keeps old content.	End of file	❌ No	open("data.txt", "a+")
'rb'	Read Binary	Opens binary file for reading.	Start	✅ Yes	open("img.png", "rb")
'wb'	Write Binary	Opens binary file for writing. Creates or overwrites.	Start	❌ No	open("img.png", "wb")
'ab'	Append Binary	Opens binary file for appending.	End	❌ No	open("img.png", "ab")
'r+b'	Read + Write Binary	Same as r+, but in binary mode.	Start	✅ Yes	open("file.bin", "r+b")
'w+b'	Write + Read Binary	Same as w+, but in binary mode.	Start	❌ No	open("file.bin", "w+b")

Mode	Full Form	Description	Pointer Position	File Must Exist?	Example
'a+b'	Append + Read Binary	Same as a+, but in binary mode.	End	❌ No	open("file.bin", "a+b")
'rt'	Read Text	Default (same as 'r').	Start	✅ Yes	open("data.txt", "rt")
'wt'	Write Text	Same as 'w', explicitly in text mode.	Start	❌ No	open("data.txt", "wt")
'at'	Append Text	Same as 'a', explicitly in text mode.	End	❌ No	open("data.txt", "at")

Modes like 'rw', 'wr', 'rr', 'ww' **do not exist in Python** — if you use them, Python will raise a ValueError.
Use the correct ones: 'r+', 'w+', 'a+', 'r+b', etc.

---

4. File Object Methods

Once a file is opened, you can use various methods to interact with it:

Method	Description	Example
read(size)	Reads specified number of characters (or entire file if not given).	file.read()
readline()	Reads one line at a time.	file.readline()
readlines()	Reads all lines and returns a list.	lines = file.readlines()
write(string)	Writes a string into the file.	file.write("Hello")
writelines(list)	Writes multiple lines (list of strings).	file.writelines(["Hi\n","Hello\n"])
seek(offset, whence)	Moves the file pointer.	file.seek(0)
tell()	Returns the current pointer position.	pos = file.tell()
flush()	Writes data from buffer to disk immediately.	file.flush()

Method	Description	Example
close()	Closes the file.	file.close()