# File Handling in Python

File handling in Python allows you to work with files — reading, writing, and manipulating data stored in text or binary formats.  
  
--------------------------------------------  
1. OPENING A FILE  
--------------------------------------------  
Python provides the built-in `open()` function to work with files.  
  
Syntax:  
 file = open("filename", "mode")  
  
Modes:  
 'r' → Read (default mode)  
 'w' → Write (overwrites existing content)  
 'a' → Append (adds new data to the end of file)  
 'x' → Create (creates a new file if not exists)  
 'b' → Binary mode (e.g., images, audio)  
 't' → Text mode (default)  
  
Example:  
 file = open("sample.txt", "r")  
  
--------------------------------------------  
2. READING A FILE  
--------------------------------------------  
You can read file content in different ways:  
  
 file.read() → Reads entire file  
 file.readline() → Reads one line  
 file.readlines() → Reads all lines as a list  
  
Example:  
 with open("data.txt", "r") as f:  
 content = f.read()  
 print(content)  
  
--------------------------------------------  
3. WRITING TO A FILE  
--------------------------------------------  
The write() and writelines() methods are used for writing.  
  
Example:  
 with open("data.txt", "w") as f:  
 f.write("Hello, Python File Handling!")  
  
Appending to existing data:  
 with open("data.txt", "a") as f:  
 f.write("\nNew line added.")  
  
--------------------------------------------  
4. WORKING WITH BINARY FILES  
--------------------------------------------  
Binary files store data such as images, videos, or audio.  
  
Example:  
 with open("image.jpg", "rb") as f:  
 data = f.read()  
  
 with open("copy.jpg", "wb") as f:  
 f.write(data)  
  
--------------------------------------------  
5. CHECKING FILE EXISTENCE  
--------------------------------------------  
You can use the `os` module to check if a file exists.  
  
Example:  
 import os  
 if os.path.exists("data.txt"):  
 print("File exists")  
 else:  
 print("File not found")  
  
--------------------------------------------  
6. FILE OPERATIONS SUMMARY  
--------------------------------------------  
- open(): Opens a file  
- read(): Reads data  
- write(): Writes data  
- close(): Closes a file  
- seek(): Moves the file cursor  
- tell(): Returns current cursor position  
  
--------------------------------------------  
7. USING 'with' STATEMENT  
--------------------------------------------  
The 'with' statement automatically closes the file after use.  
  
Example:  
 with open("test.txt", "r") as f:  
 print(f.read())  
  
--------------------------------------------  
8. EXAMPLE PROGRAM  
--------------------------------------------  
with open("example.txt", "w") as f:  
 f.write("Python is awesome!")  
  
with open("example.txt", "r") as f:  
 print(f.read())  
  
Output:  
 Python is awesome!