**Q1: Which function is used to open a file? What are the different modes of opening a file? Explain each mode of file opening.**

**Function to open a file:**  
The open() function is used to open a file in Python.

**Modes of opening a file:**

1. **r (Read mode):**
   * Opens the file for reading.
   * File must exist; otherwise, an error is raised.
   * Example: open('file.txt', 'r')
2. **w (Write mode):**
   * Opens the file for writing.
   * Creates a new file if it doesn’t exist or truncates (erases) the content if it does.
   * Example: open('file.txt', 'w')
3. **x (Exclusive creation mode):**
   * Creates a new file.
   * If the file exists, it raises a FileExistsError.
   * Example: open('file.txt', 'x')
4. **a (Append mode):**
   * Opens the file for appending new content at the end.
   * Creates a new file if it doesn’t exist.
   * Example: open('file.txt', 'a')
5. **b (Binary mode):**
   * Used to handle binary files like images and videos.
   * Combined with other modes, e.g., rb, wb.
   * Example: open('file.jpg', 'rb')
6. **t (Text mode):**
   * Default mode, used for text files.
   * Combined with other modes, e.g., rt, wt.
   * Example: open('file.txt', 'rt')
7. **r+ (Read and Write mode):**
   * Opens the file for both reading and writing.
   * File must exist.
   * Example: open('file.txt', 'r+')
8. **w+ (Write and Read mode):**
   * Opens the file for both writing and reading.
   * Truncates the file if it exists, or creates a new one.
   * Example: open('file.txt', 'w+')
9. **a+ (Append and Read mode):**
   * Opens the file for appending and reading.
   * Creates a new file if it doesn’t exist.
   * Example: open('file.txt', 'a+')

**Q2: Why is the close() function used? Why is it important to close a file?**

**Why close() is used:**  
The close() function is used to close an open file in Python. It ensures that all changes to the file are saved and that resources are released.

**Importance of closing a file:**

1. **Releases resources:** Frees up the memory and system resources used by the file.
2. **Avoids data corruption:** Ensures data is saved correctly, especially in write modes.
3. **Prevents file locks:** Closing the file prevents it from being locked by the program.
4. **Good practice:** It’s a recommended coding practice to avoid unexpected issues.

**Q3: Python program to create, write, close, and read a file.**

# Create and write to a file

file = open("data\_scientist.txt", "w")

file.write("I want to become a Data Scientist")

file.close()

# Open and read the file

file = open("data\_scientist.txt", "r")

content = file.read()

file.close()

print(content)

**Q4: Explain read(), readline(), and readlines() with examples.**

1. **read():** Reads the entire content of the file as a single string.
2. file = open("example.txt", "r")
3. content = file.read()
4. file.close()
5. print(content)
6. **readline():** Reads one line at a time from the file.
7. file = open("example.txt", "r")
8. line = file.readline()
9. file.close()
10. print(line)
11. **readlines():** Reads all lines in the file and returns them as a list of strings.
12. file = open("example.txt", "r")
13. lines = file.readlines()
14. file.close()
15. print(lines)

**Q5: Why is the with statement used with open()? What is its advantage?**

**Why with is used:**  
The with statement is used to manage resources like file handling. It ensures that the file is properly closed after its suite finishes, even if an exception occurs.

**Advantages of using with and open():**

1. **Automatic file closure:** No need to explicitly call close().
2. **Error handling:** Prevents resource leaks during exceptions.
3. **Cleaner code:** Makes the code more readable and concise.

**Example:**

with open("example.txt", "r") as file:

content = file.read()

print(content) # File is automatically closed after the block

**Q6: Explain write() and writelines() functions with examples.**

1. **write():** Writes a single string to the file.
2. with open("example.txt", "w") as file:
3. file.write("This is a single line of text.")
4. **writelines():** Writes a list of strings to the file. Each string is written as-is without adding newlines.
5. with open("example.txt", "w") as file:
6. file.writelines(["Line 1\n", "Line 2\n", "Line 3\n"])