**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.

**Different modes of opening a file:**

1. **'r' (Read Mode)**: Opens the file for reading. If the file does not exist, it raises a FileNotFoundError.
2. **'w' (Write Mode)**: Opens the file for writing. If the file exists, it overwrites the content. If the file does not exist, it creates a new file.
3. **'a' (Append Mode)**: Opens the file for appending. New data is added at the end of the file without overwriting the existing content.
4. **'x' (Exclusive Creation Mode)**: Creates a new file. If the file already exists, it raises a FileExistsError.
5. **'b' (Binary Mode)**: Opens the file in binary mode (used with other modes like 'rb' or 'wb').
6. **'t' (Text Mode)**: Opens the file in text mode (default mode).
7. **'r+' (Read and Write Mode)**: Opens the file for both reading and writing. If the file does not exist, it raises an error.
8. **'w+' (Write and Read Mode)**: Opens the file for both writing and reading. It overwrites the file if it exists; otherwise, it creates a new file.
9. **'a+' (Append and Read Mode)**: Opens the file for appending and reading. If the file does not exist, it creates a new file.

**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 a file after performing operations like reading or writing.

**Importance of closing a file:**

1. Ensures that all data is properly written to the file.
2. Frees up system resources associated with the file.
3. Prevents file corruption or data loss.
4. Makes the file available for other processes.

**Q3: Write a Python program to create a text file. Write "I want to become a Data Scientist" in that file. Then close the file. Open this file and read the content of the file.**

# Writing to the file

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

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

file.close()

# Reading the file content

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

content = file.read()

print(content)

file.close()

**Q4: Explain the following with Python code: read(), readline(), and readlines().**

1. **read()**: Reads the entire file or a specified number of characters.
2. **readline()**: Reads a single line from the file.
3. **readlines()**: Reads all lines and returns them as a list.

**Example Code:**

# Writing to a file for demonstration

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

file.write("Line 1\nLine 2\nLine 3")

# Reading using read(), readline(), and readlines()

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

print("Using read():")

print(file.read()) # Reads the entire file

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

print("\nUsing readline():")

print(file.readline()) # Reads the first line

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

print("\nUsing readlines():")

print(file.readlines()) # Reads all lines as a list

**Q5: Explain why the with statement is used with open(). What is the advantage of using with statement and open() together?**

**Why with is used with open():**  
The with statement ensures that resources, such as file handles, are properly managed. When the block of code is exited, the file is automatically closed, even if an exception occurs.

**Advantages:**

1. No need to explicitly call close()—it happens automatically.
2. Avoids resource leaks by ensuring proper cleanup.
3. Makes the code cleaner and less error-prone.

**Example:**

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

content = file.read()

print(content)

# File is automatically closed here

**Q6: Explain the write() and writelines() functions. Give a suitable example.**

1. **write()**: Writes a string to the file.
2. **writelines()**: Writes a list of strings to the file without adding newline characters automatically.

**Example:**

# Using write()

with open("write\_example.txt", "w") as file:

file.write("This is a single line.\n")

# Using writelines()

lines = ["Line 1\n", "Line 2\n", "Line 3\n"]

with open("writelines\_example.txt", "w") as file:

file.writelines(lines)