Tops Technology

# Module 15) Advance Python Programming

Presented By:

Nandni Vala

# Reading and Writing Files

1.Reading from a file using read(), readline(), readlines().

There are different methods to read data from a file: read(), readline(), and readlines(). Each method serves different purposes depending on how you want to process the file's content.

## read()

The read() method reads the entire content of the file as a single string.

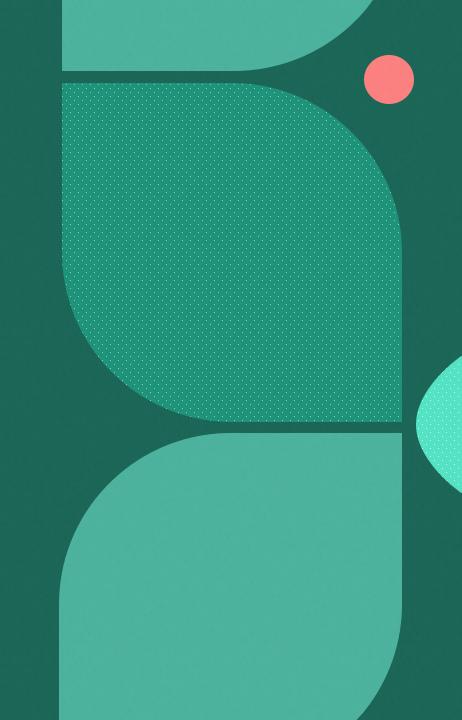
#### **Example:**

```
with open("example.txt", "r") as file:
content = file.read()
print(content)
```

# readline()

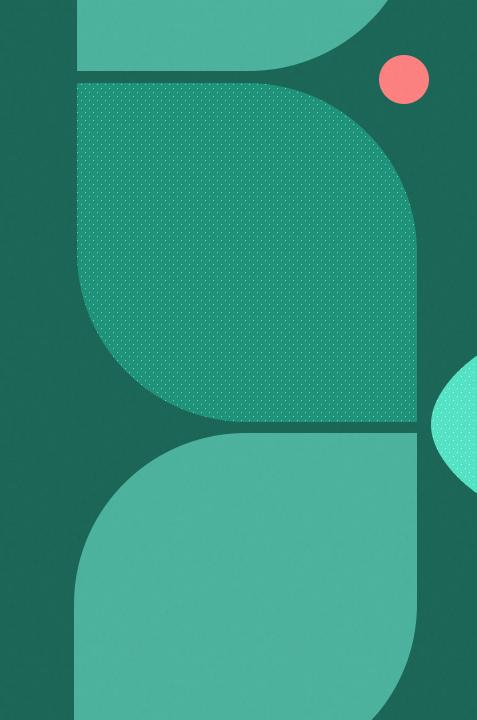
The readline() method reads one line at a time from the file. Each call reads the next line.

- > Example:
- ➤ with open("example.txt", "r") as file:
- line = file.readline()
- while line:
- print(line.strip()) # `.strip()` removes the trailing newline
- line = file.readline()
- > readlines()
- > Reads all lines from the file and returns them as a list of strings.
- ➤ Each element in the list corresponds to a single line from the file.
- **Example:**
- > with open("example.txt", "r") as file:
- lines = file.readlines()
- for line in lines:
- print(line.strip())



# 2. Writing to a file using write() and writelines().

- > write()
- ➤ Writes a single string to the file.
- ➤ Does not add a newline automatically; you must include \n if needed.
- Example:
- ➤ with open("example.txt", "w") as file:
- file.write("Hello, World!\n")
- ➤ file.write("This is another line.")
- ➤ **Result:** The file example.txt will contain:
- ➤ Hello, World!
- > This is another line.



- > writelines()
- ➤ Writes a list of strings to the file.
- ➤ Does not add newlines automatically; each string in the list should include \n if required.
- > Example:
- ➤ lines = ["Hello, World!\n", "This is another line.\n", "Goodbye!"]
- ➤ with open("example.txt", "w") as file:
- > file.writelines(lines)
- ➤ **Result:** The file example.txt will contain:
- ➤ Hello, World!
- > This is another line.
- ➤ Goodbye!