Tasks Assigned Today

19-02-2025
Suneetha.V

File Handling in Python:

File handling in Python refers to the process of reading from and writing to files. Python provides built-in functions and methods that make file operations simple.

1. Opening a File

To work with a file, you first need to open it using the `open()` function. It requires at least one argument: the name of the file. Optionally, you can specify the mode in which you want to open the file.

Syntax: file = open('filename.txt', 'r')

Common Modes:

- `'r'`: Read (default mode). Open the file for reading. The file must exist.
- `'w'`: Write. Open the file for writing. If the file exists, it will be overwritten. If it doesn’t exist, a new file is created.
- `'a'`: Append. Open the file for appending. Data is written at the end of the file, and the file is created if it doesn't exist.
- `'b'`: Binary mode. Used when working with binary files (e.g., images).
- `'x'`: Exclusive creation. Fails if the file already exists.
- `'t'`: Text mode (default mode). For working with text files (can be omitted).

2. Reading from a File

After opening the file, you can read its contents using various methods.

a. `read()`

Read the entire content of the file at once.

content = file.read()

b. `readline()`

Tasks Assigned Today

Read one line at a time from the file.

line = file.readline() # Reads the next line

c. `readlines()`

Reads all lines of the file into a list where each line is an element.

lines = file.readlines()

3. Writing to a File

You can write to a file using `write()` or `writelines()`.

a. `write()`

Writes a string to the file. If the file is opened in write or append mode, it will overwrite or append the string to the file, respectively.

file.write("Hello, world!")

b. `writelines()`

Writes a list of strings to the file. Each string in the list will be written sequentially.

lines = ["Line 1\n", "Line 2\n", "Line 3\n"] file.writelines(lines)

4. Closing the File

After you are done with file operations, you should close the file to release system resources

file.close()

5. File Existence Check

Before opening a file, you might want to check if it exists. You can use the os module to check for file existence.

```
import os

if os.path.exists("filename.txt"):

    print("File exists")
```

Tasks Assigned Today

```
else:
    print("File does not exist")
```