FILE HANDLING IN PYTHON

Overview and Practical Applications

MUKESH KUMAR

Agenda

- ✓ Introduction to File Handling
- ✓ Working with Text Files in Python
- ✓ Exception Handling in File Operations
- ✓ Using the OS Module for File Management
- ✓ Handling Large Files Efficiently
- ✓ Practical Example: Log File Handling
- ✓ Best Practices and Summary

Introduction to File Handling

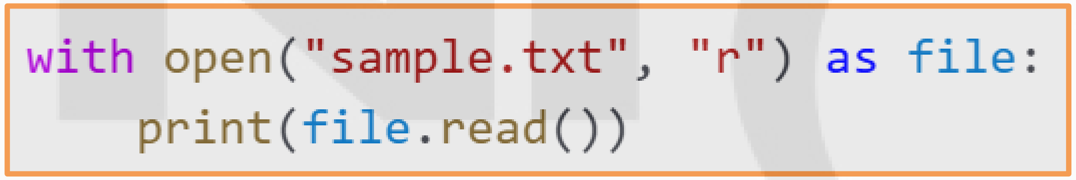
- File handling allows reading, writing, and managing files on disk.
- Python provides built-in functions to work with different file types.

Why is File Handling Important?

- Data Storage and Persistence
- Efficient Data Processing
- Automation
- Data Exchange
- Security and Backups

Working with Files

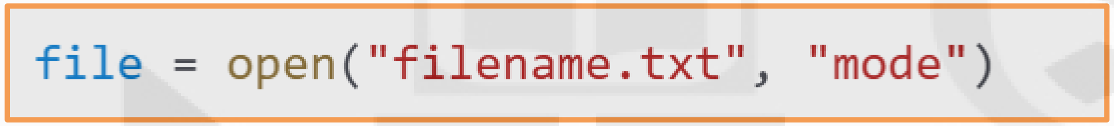
- Files that store data in human-readable format (e.g., .txt, .csv, .json)
- Example: Reading a text file



```python
with open("sample.txt", "r") as file:
    print(file.read())
```

Opening a File in Python

- Using open():



```
file = open("filename.txt", "mode")
```

- Modes:
- 'r' – Read
- 'w' – Write
- 'a' – Append
- 'x' – Create

Reading Files

- Example:

```
file = open("sample.txt", "r")
print(file.read())  # Reads entire file
file.close()

with open("sample.txt", "r") as file:
    print(file.readline())  # Reads one line
    print(file.readlines())  # Reads all lines into a list
```

Writing to Files

- Example:

```python
with open("output.txt", "w") as file:
    file.write("Hello, World!\n")
    file.writelines(["Line1\n", "Line2\n"])
```

Closing a File

- Always close files to free up resources

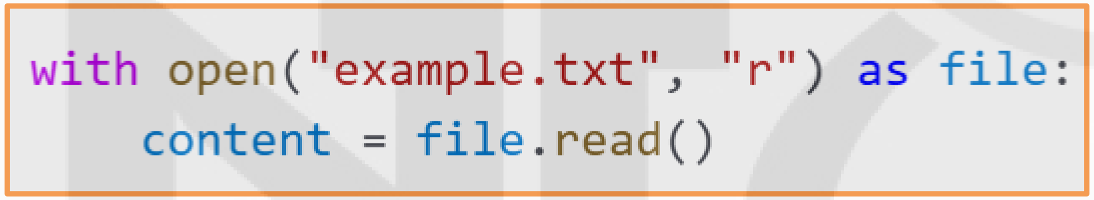
```
with open("output.txt", "w") as file:
    file.write("Hello, World!\n")
    file.writelines(["Line1\n", "Line2\n"])
```

- Best practice: Use 'with open()'

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

Handling File Exceptions

- Use try-except to handle file errors



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

- Common exceptions: FileNotFoundError, PermissionError

OS Module for File Operations

- Refer to Jupyter Notebook : OS_Module_Examples.ipynb
  - Checking if a file exists:
  - Renaming files:
  - Directory operations:
  - Reading files line-by-line to save memory:
  - Using file chunking:

Practical Example - Log File Handling

- How Python logs errors to a file:

```
import logging

logging.basicConfig(filename="app.log", level=logging.INFO)
logging.info("This is a log message")
```

- Example log entries:

```
INFO:root:This is a log message
```

Summary & Best Practices

- Always close files
- Handle exceptions
- Choose correct file modes
- Use os module for file management
- Optimize large file handling