Use Case

- **Reading and Writing Text Files**: Open, read, and write data to text files for tasks such as logging, data storage, and configuration.
- **CSV File Processing**: Read from and write to CSV files to handle tabular data for data analysis and reporting.
- **Log File Management**: Maintain and manage log files to record application events and errors.
- **Configuration Files**: Read application configuration settings from files to customize behavior without changing code.
- **Data Serialization**: Serialize and deserialize data using formats like JSON or XML to save program state or exchange data between systems.
- **Binary File Handling**: Work with binary files to read and write non-text data, such as images, videos, or executable files.
- **File Searching and Filtering**: Search for specific content within files and filter results based on certain criteria.
- **File Uploads and Downloads**: Handle file uploads and downloads in web applications to manage user files.
- **Temporary File Storage**: Create and manage temporary files to hold intermediate data during processing tasks.
- **File Compression and Decompression**: Compress and decompress files to save disk space or prepare for transfer over the network.

⬛ Creating a File

```
# Create a new file
with open("example.txt", "w") as file:
    file.write("This is a new file.")
```

⬛ Reading from a File

```
# Read the content of a file
with open("example.txt", "r") as file:
    content = file.read()
```

```
    print(content)
```

⬛ Writing to a File

```
# Write to a file
with open("example.txt", "w") as file:
    file.write("Writing new content to the file.")
```

⬛ Renaming a File

```
import os

# Rename a file
os.rename("example.txt", "new_example.txt")
```

⬛ Deleting a File

```
import os

# Delete a file
os.remove("new_example.txt")
```

⬛ Creating a Directory

```
import os

# Create a new directory
os.mkdir("new_directory")
```

⬛ Reading a Directory

```python
import os

# List all files and directories in the current directory
print(os.listdir("."))
```

⬛ Writing to a File in a Directory

```python
# Create a file in the new directory and write to it
with open("new_directory/example.txt", "w") as file:
    file.write("Writing content to a file in a new directory.")
```

⬛ Renaming a Directory

```python
import os

# Rename a directory
os.rename("new_directory", "renamed_directory")
```

⬛ Deleting a Directory

```python
import os

# Remove a file inside the directory first
os.remove("renamed_directory/example.txt")

# Delete the directory
os.rmdir("renamed_directory")
```

⬛ Read Files Name From Directory and Print List

```python
import os
directory_path = "demo"
```

```
file_list = os.listdir(directory_path)


print(file_list)

print("Files in the directory:")
for file_name in file_list:
    print(file_name)
```

⬛ Creating Zip File

```
import zipfile
with zipfile.ZipFile("my_archive.zip", "w") as zipf:
    zipf.write("example.txt")
    zipf.write("large_file.txt")
```

⬛ Extract from Zip File

```
import zipfile
zip_file_path = "my_archive.zip"
with zipfile.ZipFile(zip_file_path, "r") as zipf:
    zipf.extractall()
    extracted_files = zipf.namelist()
    print("Extracted files:", extracted_files)
```

⬛Make Zip Form Directory

```
import shutil

# Specify the directory to be zipped
directory_to_zip = "demo"

# The name of the output zip file (without extension)
output_zip_name = "demo_archive"

# Create a zip archive
shutil.make_archive(output_zip_name, 'zip', directory_to_zip)
```

```
print(f"Directory '{directory_to_zip}' has been zipped into
'{output_zip_name}.zip'")
```

⬛Make CSV File Form List

```
import csv

# Data to write
data = [
    ["Name", "Salary", "Designation", "Department", "Location"],
    ["Alice", 70000, "Software Engineer", "IT", "New York"],
    ["Bob", 85000, "Data Scientist", "Data", "San Francisco"],
    ["Charlie", 60000, "System Administrator", "IT", "Chicago"],
    ["David", 95000, "Product Manager", "Product", "Boston"],
    ["Eve", 72000, "UX Designer", "Design", "Los Angeles"]
]

# Path to the CSV file
csv_file_path = "example.csv"

# Writing to the CSV file
with open(csv_file_path, mode='w', newline='') as file:
    csv_writer = csv.writer(file)
    csv_writer.writerows(data)

print(f"CSV file '{csv_file_path}' created successfully!")
```

⬛Make List From CSV File

```
import csv

# Path to the CSV file
csv_file_path = "example.csv"

# Initialize an empty list to hold the rows
data_list = []

# Reading the CSV file and storing it in a list
with open(csv_file_path, mode='r') as file:
```

```
    csv_reader = csv.reader(file)

    # Iterate over each row in the CSV file
    for row in csv_reader:
        data_list.append(row)

# Print the list
print("List from CSV file:")
for row in data_list:
    print(row)
```