

Content Covered



- Text File
- CSV
- JSON

File Handling

- Allows user to read & write file
- Python treats file as text or binary
- File has to be closed after operation
- Context manager can be used for it
- If file is already present, write method overwrites it

Open

- Used to open file in read or write mode
- Returns file object
- Syntax: `open(file_name, mode, encoding)`
- Modes:
 - r → reading file. Default mode
 - w → writing to file
 - a → Appending

Read File

1.

```
file_obj = open('test.txt', 'r')  
for line in file_obj:  
    print(line)  
file_obj.close()
```

2.

```
file_obj = open('test.txt', 'r')  
file_content = file_obj.read() # Default read all character  
print(file_content)  
file_obj.close()
```

Write File

1.

```
file_obj = open('test2.txt', 'w')  
file_obj.write("First Line\n")  
file_obj.write("Second Line")  
file_obj.close()
```

2.

```
file_obj = open('test2.txt', 'a')  
file_obj.write("\nThird Line")  
file_obj.close()
```

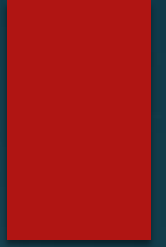
Context Manager

```
import os
```

```
file_path = os.path.join(directory, file_name)
```

```
with open(file_path, 'w') as file_obj:  
    file_obj.write("First Line\n")  
    file_obj.write("Second Line")
```

Reading CSV

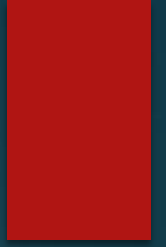


```
import csv
```

```
with open('data.csv') as file:  
    reader = csv.reader(file)  
    for row in reader:  
        print(row)
```

```
with open('data.csv') as file:  
    reader = csv.DictReader(file)  
    for row in reader:  
        print(row['Name'], row['Age'])
```

Writing CSV



```
# From List
import csv
```

```
data = [
    ['Name', 'Age', 'City'],
    ['Alice', '30', 'New York'],
    ['Bob', '25', 'Los Angeles']
]
```

```
with open('output.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerows(data)
```

Writing CSV

```
# From Dictionary  
import csv
```

```
data = [{"Name": "Alice", "Age": 30}, {"Name": "Bob", "Age": 25}]
```

```
with open("output_dict.csv", mode="w", newline="") as file:  
    headers = ["Age", "Name"] # order of field, can repeat  
    writer = csv.DictWriter(file, fieldnames= headers)  
    writer.writeheader()  
    writer.writerows(data)
```

JSON

- JavaScript Object Notation is light weight text-based format used for storing and transmitting data
- Easy to read for human and parse for machine
- Equivalent to Python dictionary
- json module is used for handling JSON files
- Example: {"Name": "Alice", "Age": 30}

Serialization

- Converting Python to json
- Converted data is of type str

Example

```
import json
```

```
data = {"Name": 'Alice', 'Age': 30, 'Skills': ['Python', 'SQL']}
```

```
json_string = json.dumps(data)  
print(json_string)
```

Writing JSON

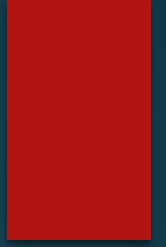
```
import json
```

```
data = {"Name": 'Alice', 'Age': 30, 'Skills': ['Python', 'SQL']}
```

```
with open('data.json', 'w') as file:
```

```
    json.dump(data, file, indent=4) # here its dump
```

Deserialization



```
import json
```

```
json_string = '{"Name": "Alice", "Age": 30}'
```

```
data = json.loads(json_string)
```

```
print(data['Age']) # Outputs: 30
```

Read JSON file

```
import json

with open('data.json') as file:
    data = json.load(file)

print(data['Name'])
```



Exercise