Python File Handling – Cheat Sheet

File Modes

- `'r'` → Read (file must exist)
- `'w'` → Write (create/overwrite file)
- `'a'` → Append (add data at end)
- `'b'` → Binary mode (images, audio)

Opening & Closing Files

```
f = open("file.txt", "r")
...
f.close()
```

Better way:

```
with open("file.txt", "r") as f:
    data = f.read()
```

(auto-closes file)

Reading Files

- `f.read()` → Entire file
- `f.readline()` → One line
- `f.readlines()` → All lines in a list

Writing Files

- `f.write("Hello")` → Write string
- `f.writelines(["a\n", "b\n"])` → Write multiple lines

Modes matter: - `'w'` → overwrite - `'a'` → append

File Paths

- **Relative path:** `open("data.txt")`
- **Absolute path:** `open("C:/Users/Name/Desktop/data.txt")`

Exception Handling

```
try:
    with open("grades.txt", "r") as f:
        print(f.read())
except FileNotFoundError:
    print("File not found!")
```

Custom Exceptions

```
class EmptyFileError(Exception):
    pass
```

Raise your own exceptions for special cases.

Best Practices

✔️Use `with` for safe file handling
✔️Handle exceptions gracefully
✔️Choose correct mode (`r`, `w`, `a`)
✔️Be careful not to overwrite files

Quick Exercises

1. Open `students.txt` → If missing, print error.
2. Ask user for favorite movies → Save & read back.
3. Diary app → Append entries to `diary.txt`.
4. Custom exception `EmptyFileError` if file is empty.
5. Funny challenge: If `apology.txt` missing → Ask user to write apology & save it.