

File handling

1) ফাইল কী এবং কেন দরকার

ফাইল হলো ডিস্কে রাখা তথ্যের একটি ইউনিট (text, image, audio, binary)। প্রোগ্রামে আমরা ফাইল পড়ি/লিখি কারণ ডেটা স্থায়ীভাবে রাখতে হয় — যেমন লগ, ইউজার ডাটা, কনফিগারেশন, CSV রিপোর্ট ইত্যাদি।

2) Python-এ ফাইল ওপেন করার মৌলিক নিয়ম

```
f = open("example.txt", "r", encoding="utf-8")
data = f.read()
f.close()
```

- `open(path, mode, encoding=...)` — ফাইল ওপেন করে একটি ফাইল অবজেক্ট দেয়।
- `f.read()` — পুরো ফাইল পড়ে।
- `f.close()` — ফাইল বন্ধ করা খুব গুরুত্বপূর্ণ (রিসোর্স রিলিজের জন্য)।

তবে `close()` ম্যানুয়ালি কল না করে context manager (with) ব্যবহার করা উত্তম — যা স্বয়ংক্রিয়ভাবে ফাইল বন্ধ করে।

3) Context manager (recommended)

```
with open("example.txt", "r", encoding="utf-8") as f:
    text = f.read()
# এখানে `f` স্বয়ংক্রিয়ভাবে closed
```

এটাই প্রায় সব ক্ষেত্রে ব্যবহার করা উচিত — exception উঠলেও ফাইল বন্ধ হবে।

4) ফাইল মোড (mode) গুলো — কী জন্য কোনটা

Mode	অর্থ
'r'	Read only (ডিফল্ট). ফাইল না থাকলে error।
'w'	Write. ফাইল থাকলে মুছে দেয়, না থাকলে নতুন করে তৈরি করে।
'a'	Append. ফাইল এ নতুন ডাটা যোগ করে (বিদ্যমান রাখা)।
'x'	Exclusive create — ফাইল না থাকলে তৈরি করে, থাকলে error দেয়।
'b'	Binary mode (e.g., 'rb', 'wb') — টেক্সট নয়, বাইনারি ডেটার জন্য।
't'	Text mode (ডিফল্ট) — 'rt' ইত্যাদি।
'+'	Read & Write (update) — 'r+', 'w+' ইত্যাদি।

উদাহরণ:

```
# নতুন ফাইলে লেখার উদাহরণ
with open("notes.txt", "w", encoding="utf-8") as f:
    f.write("Hello\n")
```

5) টেক্সট ফাইল পড়ার পদ্ধতি (উপায়-বিশেষ)

```
# পুরো টেক্সট
with open("big.txt", "r", encoding="utf-8") as f:
    all_text = f.read()

# লাইন বাই লাইন লিস্ট
with open("big.txt", "r", encoding="utf-8") as f:
    lines = f.readlines() # ছোট ফাইলের জন্য OK

# মেমরি-সাশ্রয়ী: iterator (recommended for large files)
with open("big.txt", "r", encoding="utf-8") as f:
    for line in f:
        process(line) # প্রতিটি লাইন প্রক্রিয়া করো
```

read() = পুরো ফাইল স্ট্রিং, readline() = একটি লাইন, read(n) = পরবর্তী n বাইট/চর পড়বে।

6) লেখা ও আপেন্ড (write & append)

```
# write (পুরনো ডাটা মুছে দেবে)
with open("log.txt", "w", encoding="utf-8") as f:
    f.write("নতুন লাইন\n")

# append (পুরনো ডাটা রেখে নতুন যোগ করবে)
with open("log.txt", "a", encoding="utf-8") as f:
    f.write("আরেকটা লাইন\n")
```

7) বাইনারি ফাইল (ছবি, পিডিএফ ইত্যাদি)

```
# image copy - binary mode
with open("photo.jpg", "rb") as src, open("copy.jpg", "wb") as dst:
    while True:
        chunk = src.read(1024*1024) # 1MB chunk
        if not chunk:
            break
        dst.write(chunk)
```

বাইনারি ফাইল পড়লে encoding দেবেন না এবং 'rb'/'wb' ব্যবহার করবেন।

8) CSV ফাইল (csv module) — সহজ উদাহরণ

```

import csv

# CSV পড়া
with open("sales.csv", newline='', encoding="utf-8") as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        # row হচ্ছে dict: {'Date': '2025-11-28', 'Product': 'Apple', ...}
        print(row["Date"], row["Product"], row["Price"])

# CSV লেখা
with open("out.csv", "w", newline='', encoding="utf-8") as csvfile:
    fieldnames = ["name", "age"]
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()
    writer.writerow({"name": "Arman", "age": 24})

```

CSV নিয়ে সহজ কাজের জন্য pandas অনেক দ্রুত এবং সুবিধাজনক — কিন্তু csv মডিউল stdlib-এ আছে।

9) JSON ফাইল (json module)

```

import json

data = {"name": "Arman", "scores": [90, 85]}

# write JSON
with open("data.json", "w", encoding="utf-8") as f:
    json.dump(data, f, ensure_ascii=False, indent=2)

# read JSON
with open("data.json", "r", encoding="utf-8") as f:
    obj = json.load(f)

```

ensure_ascii=False দিলে বাংলা ঠিক থাকবে; indent সুন্দর ফরম্যাট করতে সাহায্য করে।

10) ফাইল পাথ ও Pathlib (আধুনিক ও সুবিধাজনক)

```
from pathlib import Path

p = Path("data") / "sales.csv"
if p.exists():
    text = p.read_text(encoding="utf-8")
else:
    p.parent.mkdir(parents=True, exist_ok=True)
    p.write_text("Date,Product,Price,Quantity\n", encoding="utf-8")
```

pathlib পাথ অপারেশন সহজ করে এবং উইন্ডোজ/লিনাক্স পার্থক্য হ্যান্ডেল করে।

11) seek() ও tell() — ফাইল পয়েন্টার নিয়ন্ত্রণ

```
with open("example.txt", "r", encoding="utf-8") as f:
    print(f.read(10))  # প্রথম 10 অক্ষর পড়ে
    pos = f.tell()     # কারেন্ট পজিশন
    f.seek(0)          # শুরুতে ফিরে যায়
```

seek(offset, whence) — whence ডিফল্ট 0 (file start), 1 (current), 2 (end)।

12) Exception handling ও নিরাপদ ফাইল অপারেশন

```
try:
    with open("secret.txt", "r", encoding="utf-8") as f:
        data = f.read()
except FileNotFoundError:
    print("ফাইল পাওয়া যায়নি")
except PermissionError:
    print("ফাইল পড়ার অনুমতি নেই")
except Exception as e:
    print("অন্যান্য ত্রুটি:", e)
```

সবসময় সম্ভাব্য এক্সসেপশন ধরতে পারবে — বিশেষ করে ফাইল অপারেশন—Permission, Disk-full ইত্যাদি ঘটতে পারে।

13) বড় ফাইল পড়ার কৌশল (Memory efficient)

- read() দিয়ে পুরো ফাইল একসাথে পড়লে মেমরি ফিল হতে পারে। বড় ফাইলের জন্য chunk বা লাইন-আইটারেশন ব্যবহার করো।
- উদাহরণ: লাইন গণনা

```
count = 0
with open("huge.log", "r", encoding="utf-8") as f:
    for _ in f:
        count += 1
```

- অথবা বাইনারি কপি-এ আমরা chunk ব্যবহার করেছি (src.read(1024*1024))

14) ফাইল পারমিশন ও ক্লোজ করার গুরুত্ব

- যখন ফাইল খুলবে, সিস্টেমে সেই ফাইলের কিছু রিসোর্স লাগবে। তাই close() বা with জরুরি।
- অন্য প্রক্রিয়া থেকে ফাইল লক/ইউজ করা থাকলে PermissionError আসতে পারে।

15) সাধারণ ভুল ও সতর্কতা (Common pitfalls)

1. open() করে close() না করা → resource leak। (Use with.)
2. ভুল encoding → বাংলা/ইউনিকোড বশত: UnicodeDecodeError আসতে পারে। প্রায়ই encoding="utf-8" ঠিক করে দেয়।

3. 'w' মোডে ভুলবশত ওপেন করলে পুরনো ডাটা মুছে যাবে। সাবধান।
4. binary ফাইল-এ text মোড ব্যবহার করলে data corrupt হবে।
5. বড় ফাইল read() করলে OutOfMemory হতে পারে — চাক্ষে পড়ো।

16) প্র্যাকটিকাল উদাহরণ — লাইন গোনা + একটি কন্ডিশন ভিত্তিক ফাইল তৈরি

```
# huge.log থেকে শুধু ERROR লাইনগুলো আলাদা ফাইলে লিখে রাখা
from pathlib import Path

src = Path("huge.log")
dst = Path("errors.log")

with src.open("r", encoding="utf-8") as fr, dst.open("w", encoding="utf-8") as fw:
    for line in fr:
        if "ERROR" in line:
            fw.write(line)
```

17) দ্রুত চেকলিস্ট (When you work with files)

- Use with open(...) always.
- Set correct mode (r/w/a/b).
- Set encoding="utf-8" for text that may include non-ascii (বাংলা).
- For CSV/JSON use stdlib modules (csv, json) or pandas for powerful ops.
- For cross-platform paths prefer pathlib.Path.

18) প্র্যাকটিস এক্সারসাইজ (তুমি করো)

1. notes.txt-এ প্রতিবার স্ক্রিপ্ট চালালে করো—আজকের তারিখ ও একটা ভাঁজ-সহ লাইন যোগ করবে। (use a mode)
2. sales.csv থেকে প্রতিটি পণ্যের মোট বিক্রয় বের করে summary.csv এ লিখে দাও। (use csv or pandas)
3. images ফোল্ডারের সব .png ফাইল একটা zip-ফাইলে কপি করো (use zipfile or shutil.copy+pathlib`).

চাইলে আমি তোমার জন্য এই তিনটি সমস্যার কোড খুলে করে দেখিয়ে দেব — কোনটা আগে করতে চাও?