

# PYTHON CHEAT SHEET

## Tệp tin (Files): Mở, Đọc, Ghi và Xử lý lỗi

### 7.1. File Handles và Context Manager

Luôn dùng `with` để đảm bảo file tự động đóng sau khi dùng xong (tránh rò rỉ tài nguyên).

#### Các chế độ (Modes):

- `'r'`: Đọc (Read - Mặc định).
- `'w'`: Ghi (Write - Ghi đè file cũ).
- `'a'`: Ghi tiếp (Append).
- `'x'`: Tạo mới (Lỗi nếu file đã tồn tại).
- `'b'`: Nhị phân (Binary - vd: `'rb'`, `'wb'`).

```
# Standard way to open file
with open('data.txt', 'r', encoding='utf-8') as f:
    content = f.read()
# File auto-closes here
```

### 7.2. Encoding và Unicode

- Luôn chỉ định `encoding='utf-8'` khi làm việc với file văn bản (Text).
- Kiểm tra encoding nguồn để tránh lỗi `UnicodeDecodeError`.

### 7.3. Đọc file hiệu quả

#### 1. Đọc từng dòng (Tiết kiệm RAM):

```
with open('big_log.txt', 'r', encoding='utf-8') as f:
    for line in f:
        # Process one line at a time
        pass
```

#### 2. Đọc Chunk (Cho file nhị phân lớn):

```
with open('video.mp4', 'rb') as f:
    # Read 8192 bytes at a time
    for chunk in iter(lambda: f.read(8192), b''):
        # Process chunk
        pass
```

### 7.5. Xử lý lỗi (Error Handling)

Bắt các ngoại lệ phổ biến khi thao tác file để tránh crash chương trình.

```
filename = 'config.txt'

try:
    with open(filename) as f:
        data = f.read()
except FileNotFoundError:
    print('File missing!')
except PermissionError:
    print('Access denied!')
```

# PYTHON CHEAT SHEET

## Pathlib, CSV/JSON và Logging

### 7.6. Pathlib (Thao tác đường dẫn)

Dùng `pathlib` (Python 3.4+) thay vì xử lý chuỗi thủ công. Hiện đại và đa nền tảng (Windows/Linux).

```
from pathlib import Path

# Create path object
p = Path('data') / 'report.txt'

# Check existence
if p.exists() and p.is_file():
    # Read text directly
    content = p.read_text(encoding='utf-8')
```

### 7.4. Ghi file an toàn (Atomic Write)

Tránh file bị hỏng (corrupt) nếu chương trình dừng đột ngột. **Giải pháp:** Ghi vào file tạm → Đổi tên (Rename).

```
import os

tmp = 'data.tmp'
dest = 'data.final'

# 1. Write to temp file
with open(tmp, 'w') as f:
    f.write('Important Data')

# 2. Rename (Atomic operation)
os.replace(tmp, dest)
```

### 7.7. Xử lý dữ liệu chuẩn (CSV và JSON)

**CSV (Excel/Data):**

```
import csv
with open('data.csv', newline='', encoding='utf-8') as f:
    reader = csv.DictReader(f)
    for row in reader:
        print(row['name']) # Access by column
```

**JSON (Web/API):**

```
import json

# Load JSON
data = json.load(open('cfg.json', encoding='utf-8'))

# Dump JSON (ensure_ascii=False for Unicode)
json.dump(data, open('out.json', 'w'),
           ensure_ascii=False, indent=2)
```

### 7.8. Logging (Ghi nhật ký)

Dùng module `logging` thay vì `print` cho ứng dụng thực tế.

```
import logging

# Basic setup
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

logger.info('Program started')
logger.warning('Disk space low')
```