

PYTHON CHEAT SHEET

Vòng lặp (Loops) & Iteration

4.1. While Loops (Vòng lặp while)

- Dùng cho logic phụ thuộc điều kiện thay đổi.
- **Cảnh báo:** Tránh *infinite loop* (vòng lặp vô hạn). Phải có điều kiện thoát hoặc `break`.

4.2. For Loops (Duyệt phần tử)

Duyệt cơ bản:

```
for item in iterable:  
    # Xu ly item...
```

Dùng `enumerate` để lấy index:

```
for i, v in enumerate(sequence):  
    # i la index, v la gia tri
```

Ghi chú: Dùng `zip` để duyệt nhiều list cùng lúc.

4.3. Break, Continue & Else

- `break`: Thoát vòng lặp ngay.
- `continue`: Bỏ qua, sang lần lặp kế.
- `else`: Chạy khi vòng lặp kết thúc tự nhiên (không gặp `break`).

```
for x in seq:  
    if cond(x):  
        break  
    else:  
        # Chay neu khong bi break
```

4.4. Comprehensions (Viết tắt)

List Comprehension:

```
# Tao list binh phuong cac so chan  
sq = [x*x for x in range(10) if x%2==0]
```

Generator Expression (Tiết kiệm nhớ):

```
gen = (x*x for x in range(10))
```

*Mẹo: Nếu logic phức tạp, hãy viết thành hàm.

4.5. Iterators & Generators

- **Iterator:** Object có `__iter__` và `__next__`.
- **Generator:** Hàm dùng `yield`, xử lý dữ liệu lớn (Lazy).

Ví dụ Fibonacci:

```
def fib(n):  
    a, b = 0, 1  
    for _ in range(n):  
        yield a  
        a, b = b, a+b
```

4.6. Performance Tips (Hiệu năng)

- **Không sửa list** khi đang vòng lặp trên chính nó.
- **Nối chuỗi:** Dùng `'.join()` thay vì cộng `(+)`.
`s = ''.join(parts)`
- Cân nhắc `multiprocessing` nếu cần xử lý song song.