PYTHON CHEATSHEET CHƯƠNG 8

KHÁI NIỆM LIST

Danh sách (list) là **chuỗi các giá trị**, tương tự string là chuỗi ký tự.

Mỗi phần tử gọi là **element** hoặc **item**.

Phần tử trong list có thể thuộc *bất kỳ kiểu dữ liệu nào*.

Khai báo:

```
for i in range(len(mylist)):
    print(i, mylist[i])
```

LIST CÓ THỂ THAY ĐỔI ĐƯỢC

- Khác với string (immutable), list **có thể thay đổi phần tử sau khi tạo**.

numbers = [1, 2, 3]

numbers[1] = 10 # → [1, 10, 3]

- Đây là tính chất quan trọng giúp list linh hoạt trong xử lý dữ liệu.

DUYỆT LIST

Duyệt từng phần tử:

```
for item in mylist:
    print(item)
```

Duyệt bằng chỉ số:

```
for i in range(len(mylist)):
    print(i, mylist[i])
```

Kỹ thuật duyệt list được dùng trong nhiều chương trình để lọc, tính toán, tìm kiếm.

CÁC PHÉP TOÁN VỚI LIST

Một số phép toán cơ bản:

Phép toán	Ý nghĩa
+	nối list
*	lặp lại list
in	kiểm tra tồn lại
len()	số phần tử

Ví dụ:

a = [1, 2] + [3, 4] # → [1, 2, 3, 4]

b = [0] * 3 # → [0, 0, 0]

3 in [1, 2, 3] # → True

LIST SLICES, METHODS & DELETING

List slice:

```
nums = [10, 20, 30, 40]
print(nums[1:3])   # → [20, 30]
```

Xóa phần tử:

```
del nums[1]
```

LISTS & FUNCTIONS – ALIASING – PARSING LINES

List được truyền **tham chiếu**,

không phải bản sao

=> Hàm có thể thay đổi nội

dung list gốc.

```
def modify(x):
    x[0] = 99
```

Aliasing (hai biến cùng trỏ một list)

```
a = [1, 2, 3]
b = a
b[1] = 10   # a cũng thay đổi
```

Lists & Strings

Tách chuỗi thành list:

```
s = "a b c"
lst = s.split()     # → ['a','b','c']
```

Kết hợp list thành chuỗi:

```
" ".join(lst)
```