

Chapter 8:List

× □ _ Concatenating Lists Using +

□ _ Định nghĩa

nối chuỗi hoặc ghép lại) là hành động kết hợp hai hoặc nhiều phần tử lại thành một
Dùng + để tạo list mới từ 2 list có sẵn.

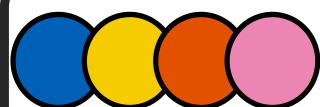
□ _ Ví dụ

```
>>> a = [1, 2, 3]
>>> b = [4, 5, 6]
>>> c = a + b
>>> print(c)
[1, 2, 3, 4, 5, 6]
```

Slicing Lists

- Dùng cú pháp list[start:stop] để lấy một phần của list.

📌 Ví dụ: a = [10, 20, 30, 40, 50]
print(a[1:4]) # [20, 30, 40]



Lists and Definite loops

Định nghĩa

Definite loop là vòng lặp mà ta biết trước số lần lặp
List là một dạng collection trong Python — cho phép lưu nhiều giá trị trong một biến duy nhất.

Ví dụ

```
friends = ['Joseph', 'Glenn', 'Sally']
for friend in friends:
    print('Happy New Year:', friend)
print('Done!')
```

List mutability

Định nghĩa

Mutability nghĩa là có thể thay đổi được sau khi tạo

Ví dụ

```
>>> lotto = [2, 14, 26, 41, 63]
>>> print(lotto)
[2, 14, 26, 41, 63]
>>> lotto[2] = 28
>>> print(lotto)
[2, 14, 28, 41, 63]
```

List

Functions

len(list) → trả về số phần tử trong list.
min(list) → trả về phần tử nhỏ nhất.
max(list) → trả về phần tử lớn nhất.
sum(list) → tính tổng các phần tử số trong list.

Methods

append(x) – thêm phần tử vào cuối list
remove(x) – xóa phần tử đầu tiên có giá trị bằng x

× □ _

List constants

□ _ Định nghĩa

- Dùng dấu ngoặc vuông [], phần tử cách nhau bằng dấu phẩy.
- Phần tử có thể là bất kỳ kiểu dữ liệu, kể cả list lồng nhau.
- Có thể tạo list rỗng: [].

□ _ Ví dụ

```
>>> print([1, 24, 76])
[1, 24, 76]
>>> print(['red', 'yellow', 'blue'])
['red', 'yellow', 'blue']
```

Lookup

Lookup là hành động tìm kiếm giá trị trong list

📌 Ví dụ fruits = ["apple", "banana", "cherry"]
print("banana" in fruits) # True

Index

- Indexing là cách truy cập phần tử trong list

📌 Ví dụ a = ["apple", "banana", "cherry"]
print(a[0]) # 'apple'

Split

- split(): Tách chuỗi thành list các từ, mặc định theo khoảng trắng.
- 📌 Ví dụ: "hello world".split() → ["hello", "world"]

