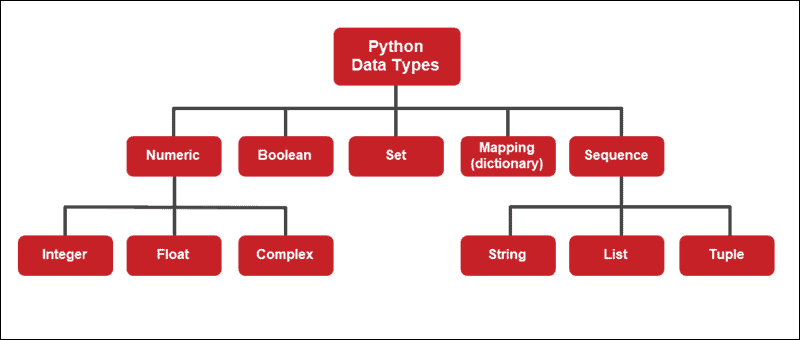
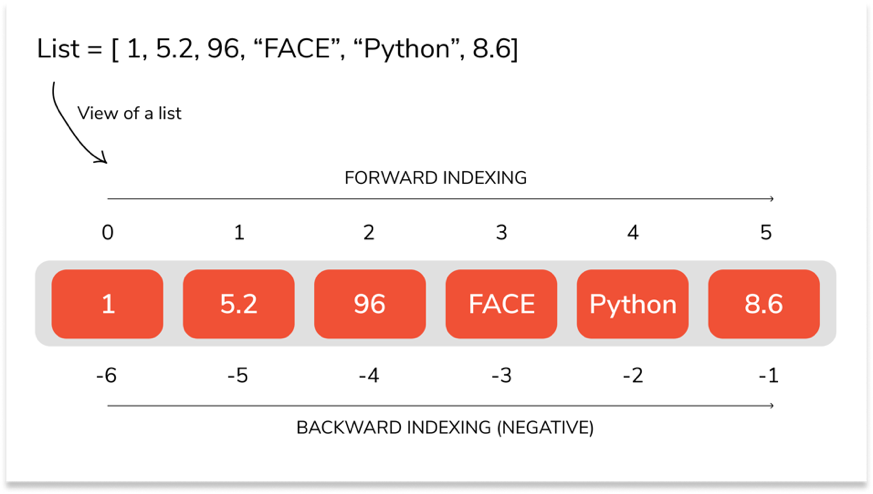
**Lab 9 : CẤU TRÚC DỮ LIỆU LIST, DICTIONARY**

****

**Sơ đồ 5 kiểu dữ liệu trong Python**

1. **HIỂU VỀ LIST, DICTINARY, TUPLE**
2. **LIST**

List dùng để lưu các giá trị (item) vào cùng một biến. Cho phép dữ liệu trùng nhau

* List được tạo ra dựa vào cặp dấu ngoặc vuông []

listStudent = ["IT001", "IT002", "IT002"]

* Được truy xuất thông qua vị trí (index)

print(listStudent[0])

print(listStudent[1])

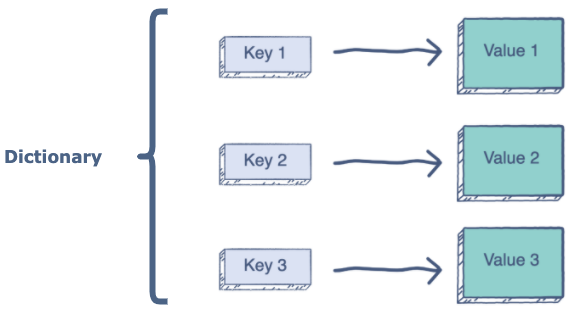
* Thay đổi giá trị tại vị trí nào đó

listStudent[2] = "IT003"

print(listStudent)

Output: ['IT001', 'IT002', 'IT003']

1. **DICTIONARY**

****

* Kiểu dictionary thường dùng để lưu giá trị 1 cặp có dạng key: value
* Không cho phép khóa key trùng giá trị
* Được bao bởi cặp dấu ngoặc nhọn { }

**Ví dụ:**

adict = {

"001": "Nguyễn Văn A",

"002": "Nguyễn Thị B",

"003": "Trần Anh C"

}

print(adict)

1. **TUPLE**

Tuple là một container cũng được sử dụng rất nhiều trong các chương trình Python không thua kém gì List

Một Tuple gồm các yếu tố sau:

* Được giới hạn bởi cặp ngoặc (), tất cả những gì nằm trong đó là những phần tử của Tuple.
* Các phần tử của Tuple được phân cách nhau ra bởi dấu phẩy (,).
* Tuple có khả năng chứa mọi giá trị, đối tượng trong Python.

1. **CẤU TRÚC CÁC HÀM LIST, DICTIONARY VÀ T VÍ DỤ**
2. **LIST**

Biến dùng chung cho các hàm bên dưới:

hoten = 'Nguyen Van Teo'

chu = ['A','B','C','D']

dayso = ['1','2','9','7','10']

1. **list(data):** Hàm này có tác dụng chuyển đổi kiểu dữ liệu của một biến sang dạng list.

**Ví dụ:**

hoten = 'Nguyen Van Teo' # gán hoten với str là Nguyen Van Teo

print(list(hoten)) # xuất ra mán hình toàn bộ ký tự trong 'hoten'

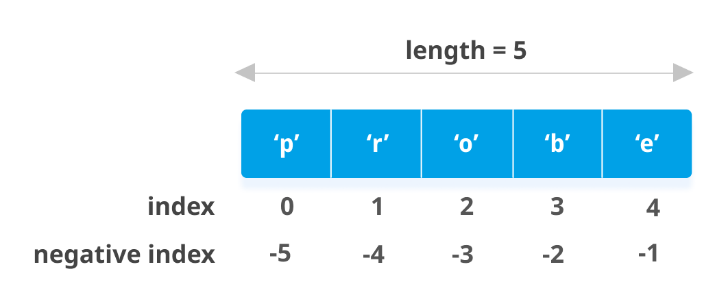
# kết quả:['N', 'g', 'u', 'y', 'e', 'n', ' ', 'V', 'a', 'n', ' ', 'T', 'e', 'o']

chu = ['A','B','C','D']

print(list(chu))

# kết quả:['A', 'B', 'C', 'D']

1. **len(data):** Hàm này trả về số lượng phần tử có trong list.

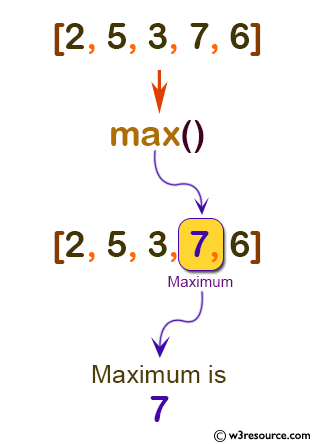


**Ví dụ:**

print(len(hoten)) # kết quả: ố lượng phần tử có trong 'hoten' là 14

print(len(chu)) # kết quả: ố lượng phần tử có trong 'chu' là 4

1. **max(data):** Hàm này sẽ trả về phần tử có giá trị lớn nhất trong list. Nếu là chuỗi thì nó sẽ trả về phần tử có độ dài chuỗi dài nhất, nếu là số thì nó sẽ trả về phần tử có số lớn nhất.



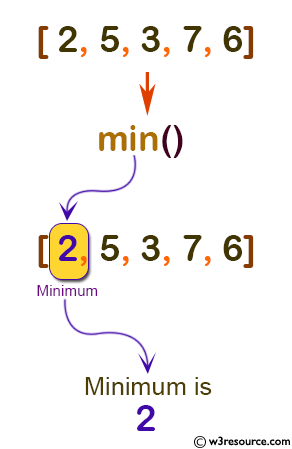
**Ví dụ:**

dayso = ['1','2','9','7','10']

print(max(hoten)) # kết quả: y là max của 'hoten'

print(max(dayso)) # trả về kết quả max là 9 mà ko phải 10 vì xem xét phần tử đơn

1. **min(data):** Hàm này sẽ trả về phần tử có giá trị nhỏ nhất trong list. Nếu là chuỗi thì nó sẽ trả về phần tử có độ dài chuỗi ngắn nhất, nếu là số thì nó sẽ trả về phần tử có số nhỏ nhất.

****

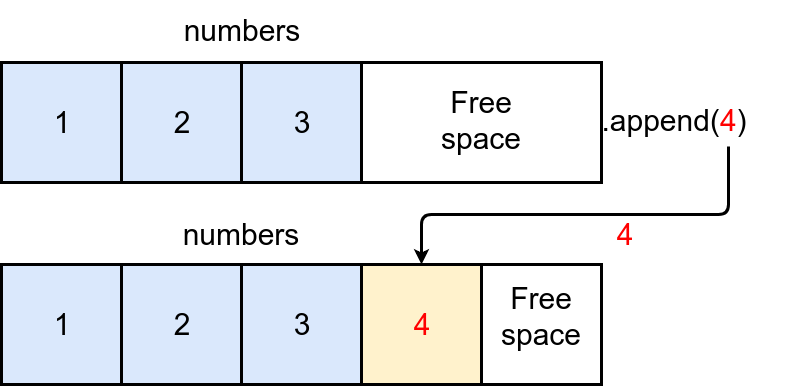
**Ví dụ:**

dayso = ['1','2','9','7','10']

print(min(chu)) # kết quả min là ký tự 'A'

print(min(dayso)) # kết quả min là '1'

1. **mylist.append(data):** Hàm này có tác dụng thêm phần vào cuối của một list.

****

**Ví dụ:**

chu.append('E') # thêm ký tự 'E' vào cuối cùng của dãy chữ

chu.append('E')

print(chu) # kết quả: ['A', 'B', 'C', 'D', 'E', 'E']

1. **mylist.remove(data):** Hàm này xóa 1 ký tự ra khỏi list

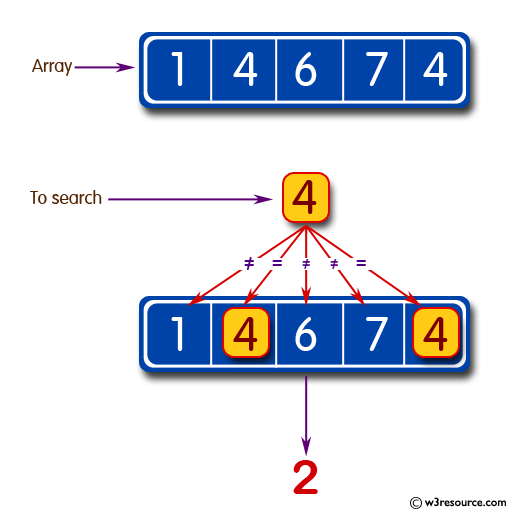


**Ví dụ:**

thislist = ["apple", "banana", "cherry"]  
thislist.pop(1)  
print(thislist)

kết quả : ['apple', 'cherry']

1. **mylist.count(val):** Hàm này có tác dụng đếm số lần xuất hiện của một thành phần trong list!



**Ví dụ:**

print(chu.count('E')) # kết quả là có 2 ký tự 'E' trong 'chu' xuất hiện

1. **DICTIONARY**
2. **Truy xuất phần tử trong Dictionary**

adict = {

"001": "Nguyễn Văn A",

"002": "Nguyễn Thị B",

"003": "Trần Anh C"

}

print(adict["002"]) #lấy value dựa vào key “002”

#hoặc dùng phương thức get()

x = adict.get("002")

print(x)

1. **Thêm phần tử**

adict["004"] = "Nguyễn D"

print(adict)

Output: {'001': 'Nguyễn Văn A', '002': 'Nguyễn Thị B', '003': 'Trần Anh C', '004': 'Nguyễn D’}

**c. Lấy danh sách các keys**

adict["004"] = "Nguyễn D"

print(adict)

Output: {'001': 'Nguyễn Văn A', '002': 'Nguyễn Thị B', '003': 'Trần Anh C', '004': 'Nguyễn D’}

**3. TUPLE (sinh vien tự tham khảo)**

1. **BÀI TẬP**

**Cho người dùng nhập vào 1 danh sách chứa các số (với số lượng phần tử do người dùng nhập)**

**Bài 1:** Viết chương trình tính tổng các phần tử trong danh sách

**Gợi ý:**

* Tạo biến tong = 0
* Duyệt từng vị trí trong danh sách
* Cộng lần lượt các giá trị lại
* return tong

**Bài 2:** Viết chương trình tính tích các phần tử trong danh sách

**Gợi ý:**

* Tạo biến tich = 1
* Duyệt từng vị trí trong mảng
* Nhân lần lượt các giá trị lại
* return tich

**Bài 3:** Viết chương trình tìm giá trị lớn nhất các phần tử trong danh sách

**Gợi ý:**

* Tạo biến max = phần tử đầu tiên của list
* Duyệt từng vị trí trong mảng
* So sánh các giá trị với phần tử max
* return max

**Bài 4:** Viết chương trình tìm giá trị nhỏ nhất các phần tử trong danh sách

**Gợi ý:**

* Tạo biến min = phần tử đầu tiên của list
* Duyệt từng vị trí trong mảng
* So sánh các giá trị với phần tử min
* return min

**Bài 5:** Viết chương trình in ra các giá trị là số lẻ các phần tử trong danh sách

**Gợi ý:**

* Tạo 1 list rỗng result = []
* Duyệt từng vị trí trong mảng chính
* Kiểm tra điều kiện nếu số nào chia hết cho 2 khác 0.
* Thì thêm phần tử vào return result

**Bài 6:** Viết chương trình in ra các giá trị là số chẵn các phần tử trong danh sách

**Gợi ý:**

* Tạo 1 list rỗng result = []
* Duyệt từng vị trí trong mảng chính
* Kiểm tra điều kiện nếu số nào chia hết cho 2 bằng 0. Thì thêm phần tử vào
* return result

**Bài 7:** Bạn được cung cấp một danh sách N bài hát đã từng được nghe của một người dùng ứng dụng Zingmp3.

Giả sử bạn không biết trong danh sách đó có tồn tại bài hát nào được lặp lại hay không. Yêu cầu đặt ra là tạo ra một danh sách mới mà không có bài hát nào được lặp lại (loại bỏ phần tử trùng trong danh sách)

**Gợi ý:**

* Input: Tự tạo ra một danh sách chứa tên bài hát mà các phần tử trong đó có thể trùng tên nhau
* Output: Một danh sách khác mà trong đó không có bài nào trùng

**Bài 8:** Cho người dùng nhập một từ điển tiếng Anh (khoảng 10 từ) với từ vựng là key và nội dung là phần diễn giải nghĩa của từ đó.

**Gợi ý:**

* Dùng dictionary
* Yêu cầu người dùng nhập vào từ cần tra. Nếu từ đó có trong từ điển thì hiển thị nội dung. Ngược lại thông báo không tìm thấy

**­­­­­­­­­­**