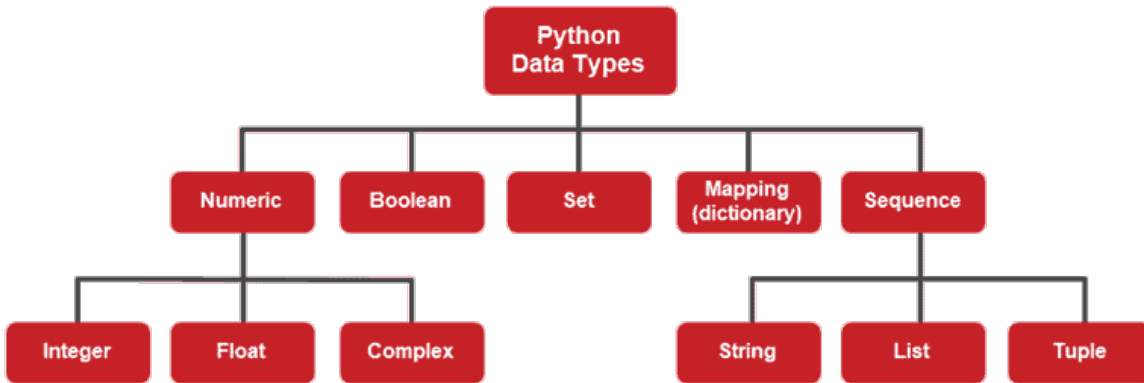




Lab 9 : CẤU TRÚC DỮ LIỆU LIST, DICTIONARY, TUPLE

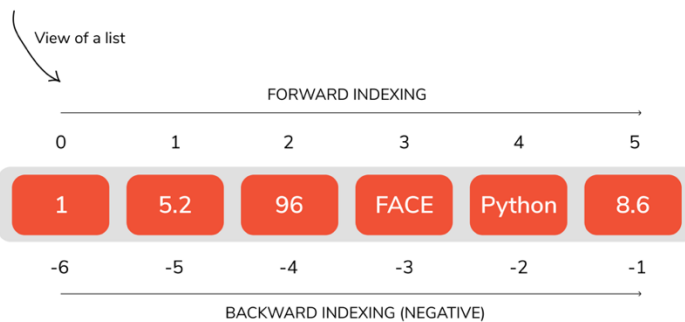


Sơ đồ 5 kiểu dữ liệu trong Python

A. HIỂU VỀ LIST, DICTIONARY, TUPLE

1. LIST

List = [1, 5.2, 96, "FACE", "Python", 8.6]



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

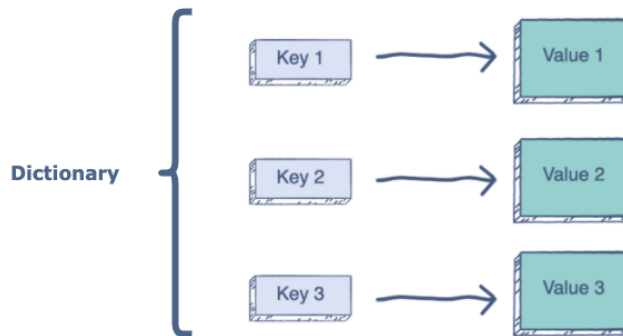
```
listStudent[2] = "IT003"
```

```
print(listStudent)
```



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

2. 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)
```

3. 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.

B. CẤU TRÚC CÁC HÀM LIST, DICTIONARY VÀ T VÍ DỤ

1. LIST

Biến dùng chung cho các hàm bên dưới:

```
hoten = 'Nguyen Van Teo'
```



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

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

- a. **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 màn 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']
```

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

	length = 5				
index	0	1	2	3	4
negative index	-5	-4	-3	-2	-1

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
```

- c. **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.



[2, 5, 3, 7, 6]

↓
max()

[2, 5, 3, **7**, 6]

Maximum

Maximum is
7

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
```

- d. 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.

[2, 5, 3, 7, 6]

↓
min()

[**2**, 5, 3, 7, 6]

Minimum

Minimum is
2

Ví dụ:

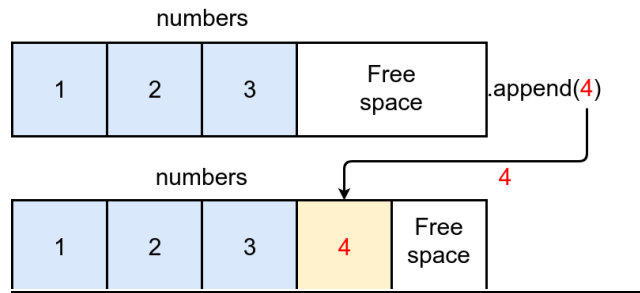
```
dayso = ['1','2','9','7','10']
```

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

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



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



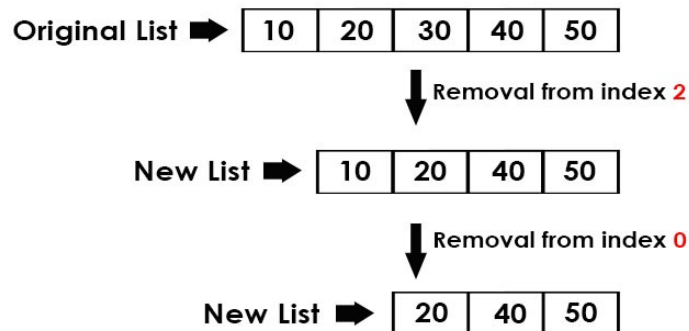
Ví dụ:

`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']

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



Ví dụ:

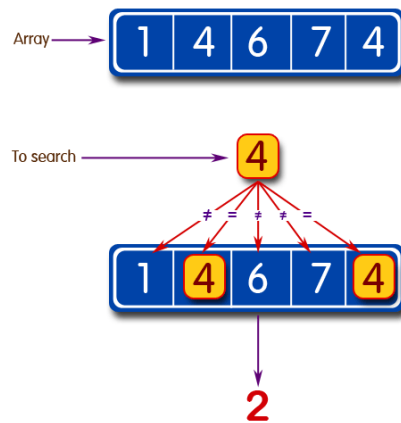
`thislist = ["apple", "banana", "cherry"]`

`thislist.pop(1)`

`print(thislist)`

kết quả : ['apple', 'cherry']

g. **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
```

2. DICTIONARY**a. 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)
```

b. 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 viên tự tham khảo)

C. BÀI TẬP

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

Gợi ý:

- Tạo biến tong = 0
- Duyệt từng vị trí trong mảng
- 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 tích = 1
- Duyệt từng vị trí trong mảng
- Nhân lần lượt các giá trị lại
- return tích

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 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 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.

Bạn cần xây dựng từ danh sách trên một chuỗi dài nhất các bài hát liên tiếp trong đó không có bài hát nào được lặp lại

Gợi ý:

- Input: A – Mảng chứa id bài hát
- Output: Độ dài cần tìm.
- Example: Input: A = {1,2,1,3,2,7,4,2,5,5}
- Output: 6 (vì chuỗi hình thành được là: {1,2,3,7,4,5})