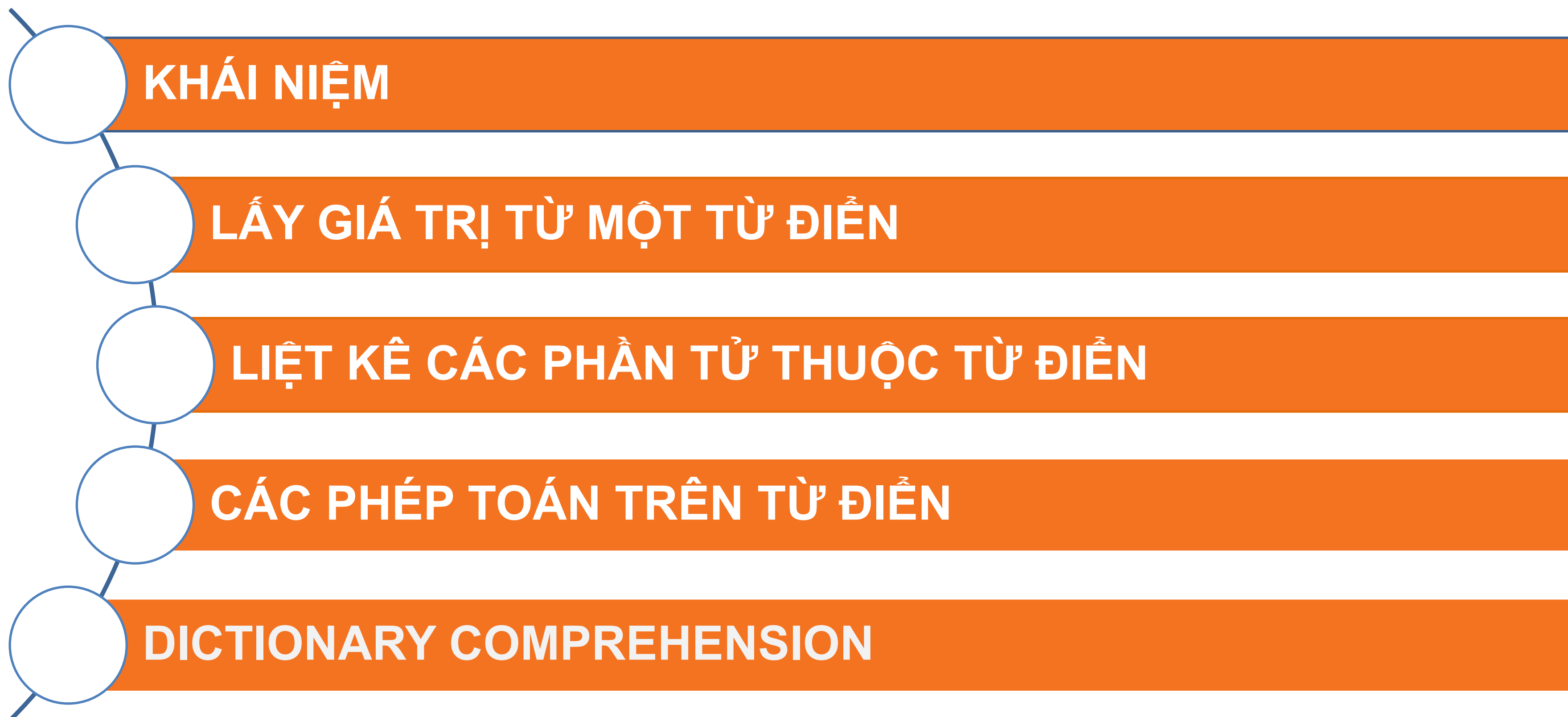




LẬP TRÌNH PYTHON

Chương 9 – Dictionaries

NỘI DUNG



- Từ điển (dict) là một danh sách các phần tử, mỗi phần tử gồm hai thành phần, một thành phần được gọi là khóa (key), thành phần còn lại gọi là giá trị (value) của thành phần đó.
- Các phần tử trong từ điển có khóa khác nhau, còn giá trị thì có thể bằng nhau.
- Khóa của mỗi phần tử không thay đổi được, còn giá trị của nó thì có thể thay đổi.

Khai báo:

< biến từ điển > = { <khóa 0>:<giá trị 0>, <khóa 1>:<giá trị 1>, .. , <khóa n-1>:<giá trị n-1> }

Ví dụ:

`Dict = {1: 2, 2: "Toán", 3: "Tin"}`

LẤY MỘT GIÁ TRỊ TỪ MỘT TỪ ĐIỂN

- Các phần tử trong từ điển không được đánh chỉ số như kiểu dữ liệu danh sách, các phần tử trong từ điển được xác định thông qua khóa của chúng
- Đưa ra giá trị của một phần tử thông qua khóa của nó

Cú pháp:

<biến từ điển>[khóa]

```
Dict = {1: 2, 2: "Toán", 3: "Tin"}
```

```
print(Dict[1])
```

LẤY MỘT GIÁ TRỊ TỪ MỘT TỪ ĐIỂN SỬ DỤNG GET()

➤ Phương thức get() không tham số:

- Trả về giá trị của phần tử có khóa là k. Trong trường hợp, từ điển không chứa phần tử có khóa k, phương thức trả về giá trị **None**:
- Cú pháp: **Dict.get(k)**

➤ Phương thức get() dạng tham số:

- Trả về giá trị của phần tử có khóa là k. Trong trường hợp, từ điển không chứa phần tử có khóa k, phương thức trả về giá trị “giá trị mặc định”.
- Cú pháp: **Dict.get(k, “giá trị mặc định”)**

```
Dict = {1: 2, 2: "Toan", 3: "Tin", 4: "Ly"}
```

```
print(Dict.get(2))
```

```
print(Dict.get(5, "Khong ton tai"))
```

KIỂM TRA SỰ TỒN TẠI CỦA KHOÁ

- Nếu khoá không tồn tại, ngoại lệ **KeyError** sẽ xuất hiện
- Dùng toán tử **in** và **not in** để kiểm tra khoá liệu có tồn tại trong từ điển

```
Dict = {1: 2, 2: "Toán", 3: "Tin"}
```

```
print(2 in Dict)
```

```
print(3 not in Dict)
```

LIỆT KÊ CÁC PHẦN TỬ THUỘC TỪ ĐIỂN

➤ Liệt kê các khoá

- *Cú pháp minh họa:*

```
for x in Dict:
```

➤ Liệt kê các giá trị

- *Cú pháp minh họa:*

```
for x in Dict:
```

```
    print(Dict[x], end = ", ")
```

*Hoặc dùng phương thức **value()***

```
for x in Dict.values():
```

```
    print(x, end = ", ")
```


LIỆT KÊ CÁC PHẦN TỬ THUỘC TỪ ĐIỂN

➤ Liệt kê cả khoá và giá trị:

- Để liệt kê cả khóa và giá trị của các phần tử trong từ điển, ta sử dụng phương thức **items()**.
- *Cú pháp minh họa:*

```
for x, y in Dict.items():  
    print(x, '::', y)
```


➤ **Lấy số phần tử thuộc từ điển:**

- Sử dụng hàm `len(<biến từ điển>)`
- *Cú pháp minh họa:*

```
Dict = {1: 2, 2: "Toan", 3: "Tin" }
```

```
print("Số phần tử của từ điển: ", len(Dict))
```

➤ Thêm một phần tử vào từ điển:

- Từ điển là đối tượng có thể thay đổi
- Để thêm cặp khóa-giá trị mới

<biến từ điển> [<khoá mới>] = <giá trị mới>

```
Dict = {1: 2, 2: "Toán", 3: "Tin"}
```

```
Dict['key4'] = "C++"
```

```
print(Dict['key4'])
```

➤ **Thay đổi giá trị một phần tử:**

<biến từ điển> [<khoá>] = <giá trị mới>

```
Dict = {1: 2, 2: "Toán", 3: "Tin"}
```

```
Dict[3] = "Web"
```

```
print(Dict[3])
```

➤ Xoá một phần tử thuộc từ điển:

❖ Phương thức **pop()**:

- Xoá phần tử có khóa <khóa> khỏi từ điển: Dict.**pop**(<khóa>)

❖ Phương thức **popitem()**:

- Xoá phần tử được đưa vào từ điển lần sau cùng: Dict.**popitem**()

❖ Hàm **del**:

- Xoá phần tử có khóa <khóa> khỏi từ điển: **del** Dict[<khóa>]
- Xoá cả từ điển: del Dict

❖ Phương thức **clear()**:

- Xoá tất cả các phần tử thuộc từ điển: Dict.**clear**()

Dict là tên biến từ điển

THÊM CÁC PHẦN TỬ VÀO MỘT TỪ ĐIỂN HIỆN CÓ

- Từ điển là đối tượng có thể thay đổi
- Để thêm cặp khóa-giá trị mới

dictionary[key] = value

```
dict_val = {1: 2, 2: "Toán", 3: "Tin"}
```

```
dict_val['key4'] = "C++"
```

```
dict_val[3] = "Web"
```

```
print(dict_val['key4'])
```

```
print(dict_val[3])
```

Cú pháp:

{**biểu_thức_khóa**: **biểu_thức_giá_trị** **for** phần_tử **in** tập_hợp}

```
dict = {x: x*2 for x in range(5)}  
print(dict)
```

#output:

```
{0: 0, 1: 2, 2: 4, 3: 6, 4: 8}
```

- Nắm được khái niệm Dictionary
- Thực hiện lấy được một giá trị và liệt kê các phần tử thuộc từ điển
- Vận dụng các phép toán trong Dictionary để giải quyết các bài toán



HỎI ĐÁP





Trân trọng cảm ơn!