ĐỀ CƯƠNG ÔN TẬP HK1 – KHỐI 11

CÂU LỆNH XUẤT:

Cú pháp:

```
print(<THÔNG TIN 1>,< THÔNG TIN 2>,…,< THÔNG TIN n>)
```

Tác dụng:

Đưa các các THÔNG TIN ra màn hình.

THÔNG TIN có thể là:

- "Chuỗi" ('Chuỗi')
- Giá trị cụ thể
- Biến(giá trị của biến)
- Biểu thức(Kết quả của biểu thức)

CÂU LỆNH NHẬP:

Cú pháp:

```
<Tên biến> = input('Chuoi thông báo nhập')
```

Tác dụng: Dùng để lưu một xâu kí tự được nhập từ bàn phím vào <Tên biến>.

Lưu ý: *Mặc định giá trị được nhập vào từ bàn phím sẽ là kiểu chuỗi*

Ép kiểu:

Để nhận các kiểu dữ liệu khác nhau, khi nhập dữ liệu chúng ta sẽ tiến hành ép kiểu.

KIỂU DỮ LIỆU	TÊN HÀM	VÍ DỤ
Số nguyên	int(n)	int(5.5) → 5
Số thực	float(n)	float(5) → 5.0
Chuỗi	str(n)	str(5) → "5"
Logic	bool(n)	bool('true') → True

Lưu ý: *Các hàm ép kiểu chỉ thực hiện được khi dữ liệu có dạng tương tự kiểu định ép.*

Ví dụ: **int("Vi du")** không thể thực hiện được

CÂU LỆNH GÁN

Cú pháp:

<Tên biến> = <Giá trị>

Lưu ý:

- <Tên biến> Bên trái bắt buộc là biến
- <Giá trị> Có thể là biến, giá trị cụ thể hoặc biểu thức.

Lệnh cộng dồn:

a = a + i → a += i

Dạng bài tập: Cho các câu lệnh gán, xác định giá trị của biến thay đổi như thế nào sau khi thực hiện các câu lệnh gán

BIỂU THỨC ĐIỀU KIỆN:

Các toán tử:

+	-	*	/	//	%
Cộng	Trừ	Nhân	Chia	Chia lấy nguyên	Chia lấy dư

Các phép so sánh:

>	<	>=	<=	==	!=
Lớn hơn	Nhỏ hơn	Lớn hơn hoặc bằng	Nhỏ hơn hoặc bằng	Bằng (Không nhầm với phép gán)	Khác

→ kết quả trả ra là kiểu LOGIC

Các phép quan hệ:

and	or	not
Và	Hoặc	Phủ định

→ kết quả trả ra là kiểu LOGIC

Python hỗ trợ dạng so sánh kép như

```
x = 2
1<x<3          # True
3 > x <=2      # True
2==x<4         # True
10 <x<20       # False
```

Dạng bài tập:

- Cho yêu cầu, xác định biểu thức điều kiện đúng
- Xác định kết quả của biểu thức điều kiện

CẤU TRÚC RẼ NHÁNH IF:

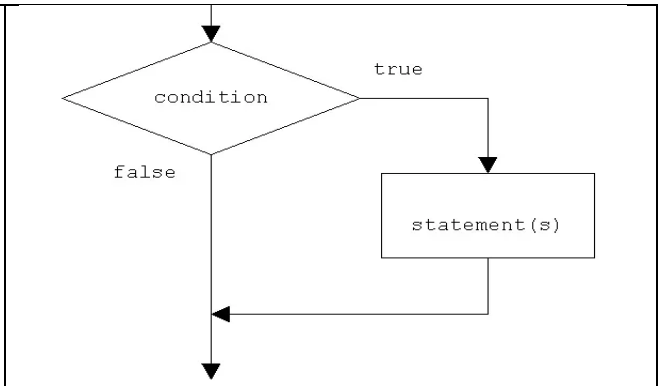
Cú pháp:

Câu lệnh if dạng thiếu:

```
if  <biểu thức điều kiện> :
                    <Khối lệnh>
```

Ý NGHĨA THỰC HIỆN

Nếu <biểu thức điều kiện> đúng thì sẽ thực hiện <Khối lệnh>

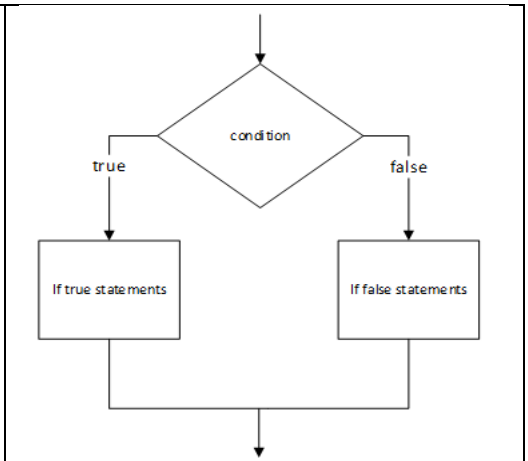


Câu lệnh if dạng đủ:

```
if  <biểu thức điều kiện> :
      <Khối lệnh 1>
else:
      <Khối lệnh 2>
```

Ý NGHĨA THỰC HIỆN:

- Nếu <biểu thức điều kiện> **ĐÚNG** thì sẽ thực hiện <Khối lệnh 1>, bỏ qua <khối lệnh 2>
- Nếu <biểu thức điều kiện> **SAI** thì sẽ thực hiện <Khối lệnh 2> bỏ qua <khối lệnh 1>



Lưu ý:

- <Khối lệnh> là 1 hoặc nhiều lệnh
- Biểu thức điều kiện cho kết quả Logic (True/False)
- Trong câu lệnh if thiếu, <Khối lệnh> có thể không được thực hiện khi …
- Trong câu lệnh if đủ, sẽ thực hiện 1 trong 2 khối lệnh

  <Khối lệnh 1> chỉ được thực hiện khi <biểu thức điều kiện> là True

  <Khối lệnh 2> chỉ được thực hiện khi <biểu thức điều kiện> là False

CẤU TRÚC LẶP WHILE:

Cú pháp:

```
while <điều_kiện> :
    <khối lệnh cần lặp>
```

1. **<khối lệnh cần lặp>** chỉ thực hiện khi điều kiện là **True**
2. **while** sẽ không lặp lần nào khi ngay lần lặp đầu tiên **ĐIỀU_KIỆN** đã **False**.
3. **while** sẽ **thực hiện mãi mãi** (vòng lặp vô hạn) khi **ĐIỀU_KIỆN** luôn bằng **True**.

➔Khi sử dụng câu lệnh **WHILE** để **điều khiển lặp**, ta cần chú ý đến:

- **Câu lệnh khởi tạo điều kiện?**
- **Biểu thức điều kiện?**
- **Câu lệnh cập nhật điều kiện?**

Dạng bài tập:

- Xác định điều kiện lặp.
- Số lần thực hiện `<khối lệnh cần lặp>`

CẤU TRÚC LẶP FOR:

Cú pháp:

```
for  <biến chạy> in  <tập các giá trị> :
                <khối lệnh cần lặp>
```

Thực hiện:

Để tạo ra vòng lặp, **<BIẾN CHẠY>** sẽ **tự động** nhận từng giá trị có trong **<TẬP CÁC GIÁ TRỊ>**.

Mỗi lần **<BIẾN CHẠY>** nhận 1 giá trị → thực hiện **<KHỐI LỆNH CẦN LẶP>** 1 lần . ***Như vậy số lần lặp bằng số phần tử có trong*** *<TẬP CÁC GIÁ TRỊ>*

<TẬP CÁC GIÁ TRỊ>: có thể là một danh sách (list), chuỗi ký tự (string) hoặc các kiểu dữ liệu khác như tuple, set, …

công dụng, tên gọi, cú pháp, cách hoạt động

HÀM RANGE (DÙNG KẾT HỢP VỚI VÒNG LẶP FOR):

Cú pháp:

```
for < biến chạy> in range(<bắt đầu>,<kết thúc>,<bước nhảy>):
    <khối lệnh cần lặp>
```

Hàm range gồm:

<BẮT ĐẦU> : là **giá trị khởi gán ban đầu cho biến chạy**. Mặc định nếu không truyền tham số, giá trị bắt đầu sẽ được gán bằng 0

<KẾT THÚC> : là **giá trị kết thúc cho biến** *chạy nhưng không bao gồm chính nó* **(không bằng).**

<BƯỚC NHẢY> : là **giá trị mà biến nhảy tự động tăng sau mỗi lần lặp**. Mặc định nếu không có thì ngầm hiểu là 1.

Dạng bài tập:

- Các cách viết khác của hàm **range**
- Xác định số lần lặp của **<khối lệnh cần lặp>** thông qua biến chạy
- Xác định tập giá trị để lặp số lần tương ứng với yêu cầu