

LAB 2: Kiểu dữ liệu, Cấu trúc điều kiện, Hàm trong Python

a. Các kiểu dữ liệu cơ bản trong Python:

- **Kiểu *int*:** Kiểu số nguyên (không có chứa dấu chấm thập phân), có thể lưu các số nguyên âm và dương.
Ví dụ: 123, -24
- **Kiểu *float*:** Kiểu số thực (có chứa dấu chấm thập phân),
ví dụ: 3.5, -4.6
- **Kiểu *complex*:** Kiểu số phức,
ví dụ 1: $z = 2+3j$ thì 2 là phần thực, 3 là phần ảo (j là từ khóa để đánh dấu phần ảo)
ví dụ 2: `z=complex(2,3)` thì 2 là phần thực, 3 là phần ảo,
khi xuất kết quả ta có thể xuất:
`print("Phần thực=",z.real) → Phần thực= 2`
`print("Phần ảo=",z.imag) → Phần ảo= 3`
- **Kiểu *str*:** Kiểu chuỗi, để trong nháy đôi hoặc nháy đơn
Ví dụ: "Khoa học", 'Dữ liệu'
- **Kiểu *bool*:** Kiểu luận lý, để lưu True hoặc False
Ví dụ 1: `t1=True`
Ví dụ 2: `t2=False`

b. Cấu trúc rẽ nhánh:

❖ if

Đây là ví dụ về câu lệnh if cơ bản nhất. **Nếu ... thì ...**

- Nếu $m - 1 < 0$ thì $m < 1$

Từ đó, Python đã xây dựng một cấu trúc nếu tương tự như trên:

```
if expression:  
    # If-block
```

Ví dụ:

```
if a - 1 < 0: # (a - 1 < 0) có giá trị là True  
    print('a nhỏ hơn 1')
```

❖ If – else if

Đây là bản nâng cấp của cấu trúc **if** vừa rồi.

Nó có cấu trúc như sau:

```
if expression:
    # If-block
elif 2-expression:
    # 2-if-block
elif 3-expression:
    # 3-if-block
...
elif n-expression:
    # n-if-block
```

Ví dụ:

```
a = 3
if a - 1 < 0: # False, tiếp tục
    print('a nhỏ hơn 1')
elif a - 2 < 0: # False, tiếp tục
    print('a nhỏ hơn 2')
elif a - 3 < 0: # False, tiếp tục
    print('a nhỏ hơn 3')
elif a - 4 < 0: # True, kết thúc
    print('a nhỏ hơn 4')
elif a - 5 < 0: # Khối BIG đã kết thúc, dù đây là True nhưng không
    ý nghĩa
    print('a nhỏ hơn 5')
```

c. Hàm:

Khi định nghĩa các hàm để cung cấp một tính năng nào đó, cần theo các qui tắc sau:

- Từ khóa **def** được sử dụng để bắt đầu phân định nghĩa hàm. **Def** xác định phần bắt đầu của khối hàm.
- **def** được theo sau bởi **ten_ham** được theo sau bởi các dấu ngoặc đơn ().
- Các tham số được truyền vào bên trong các dấu ngoặc đơn. Ở cuối là dấu hai chấm.
- Trước khi viết một code, một độ thụt dòng được cung cấp trước mỗi lệnh.
- Độ thụt dòng này nên giống nhau cho tất cả các lệnh bên trong hàm đó.
- Lệnh đầu tiên của hàm là tùy ý, và nó là **Documentation String** của một hàm đó.
- Sau đó là lệnh để được thực thi.

Cú pháp hàm trong Python

```
def ten_ham( cac_tham_so ):
    "function_docstring"
    function_suite
    return [bieu_thuc]
```

Để thực thi một hàm, cần gọi hàm đó. Phần định nghĩa hàm cung cấp thông tin về tên hàm các tham số và định nghĩa những hoạt động nào được thực hiện bởi hàm đó.

Ví dụ:

```
# Phan dinh nghia ham
def printFunc( str ):
    "Chuoi nay duoc truyen vao trong ham"
    print str
    return;# Bay gio ban co the goi ham printFunc
printFunc("Loi goi dau tien toi custom func!")
```

Thực hành:

Câu 1.

Nhập 3 cạnh a, b và c của một tam giác. Hãy kiểm tra và chỉ ra hình dạng của tam giác ABC (vuông, đều, cân, tù, nhọn,...)

Câu 2.

Viết chương trình nhập số A và kiểm tra xem A có phải là số nguyên tố hay không?

Câu 3:

Viết chương trình liệt kê tất cả các số nguyên tố nhỏ hơn n. Số nguyên dương n được nhập từ bàn phím.
