

Day 4: Functions

thanhha.hust@gmail.com

3.1 Lời gọi hàm

- ❖ Một hàm được đặt tên gồm một nhóm các câu lệnh nhằm thực hiện một nhiệm vụ tính toán cụ thể

```
>>> type(32)
<type 'int'>
```

- ❖ Biểu thức ở trong cặp ngoặc đơn được gọi là đối số của hàm

3.2 Các hàm chuyển đổi kiểu

- ❖ Python cung cấp các hàm dựng sẵn giúp chuyển đổi một giá trị từ kiểu này sang kiểu khác.

```
>>> int('32')
32
>>> int('Hello')
ValueError: invalid literal for int(): Hello
>>> int(3.99999)
3
>>> int(-2.3)
-2
>>> float(32)
32.0
>>> float('3.14159')
3.14159
>>> str(32)
'32'
>>> str(3.14159)
'3.14159'
```

3.3 Các hàm toán học

- ❖ Một module là một file trong đó có tập hợp các hàm liên quan với nhau
- ❖ Python có một module (mô-đun) toán cung cấp phần lớn các hàm toán học thông dụng.

```
>>> import math
```

```
>>> print math  
<module 'math' (built-in)>
```

```
>>> ratio = signal_power / noise_power  
>>> decibels = 10 * math.log10(ratio)  
>>> radians = 0.7  
>>> height = math.sin(radians)
```

```
>>> degrees = 45  
>>> radians = degrees / 360.0 * 2 * math.pi  
>>> math.sin(radians)  
0.707106781187
```

```
>>> math.sqrt(2) / 2.0  
0.707106781187
```

3.4 Thêm một hàm mới

Tạo mới một hàm

```
def print_lyrics():  
    print "I'm a lumberjack, and I'm okay."  
    print "I sleep all night and I work all day."
```

```
>>> print print_lyrics  
<function print_lyrics at 0xb7e99e9c>  
>>> type(print_lyrics)  
<type 'function'>
```

Một khi bạn đã định nghĩa hàm, bạn có thể dùng nó trong một hàm khác

```
def repeat_lyrics():  
    print_lyrics()  
    print_lyrics()
```

```
>>> repeat_lyrics()  
I'm a lumberjack, and I'm okay.  
I sleep all night and I work all day.  
I'm a lumberjack, and I'm okay.  
I sleep all night and I work all day.
```

3.4 Thêm một hàm mới

❖ Hàm với tham số mặc định:

```
>>> def cong(a,b=2):  
...     print a+b  
...  
>>> cong(2)  
4  
>>> cong(2,3)  
5
```

Chú ý: Tham số của hàm được gán giá trị mặc định phải sau tham số không được gán giá trị mặc định

```
>>> def cong(a,b=2,c=3):  
...     print a+b+c  
...  
>>> def cong(a=2,b,c=3):  
...     print a  
...  
File "<stdin>", line 1  
SyntaxError: non-default argument follows default argument
```

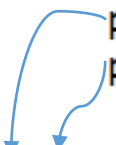
3.4 Thêm một hàm mới

Error!!!

```
def repeat_lyrics():
    print_lyrics()
    print_lyrics()

def print_lyrics():
    print "I'm a lumberjack, and I'm okay."
    print "I sleep all night and I work all day."

repeat_lyrics()
```



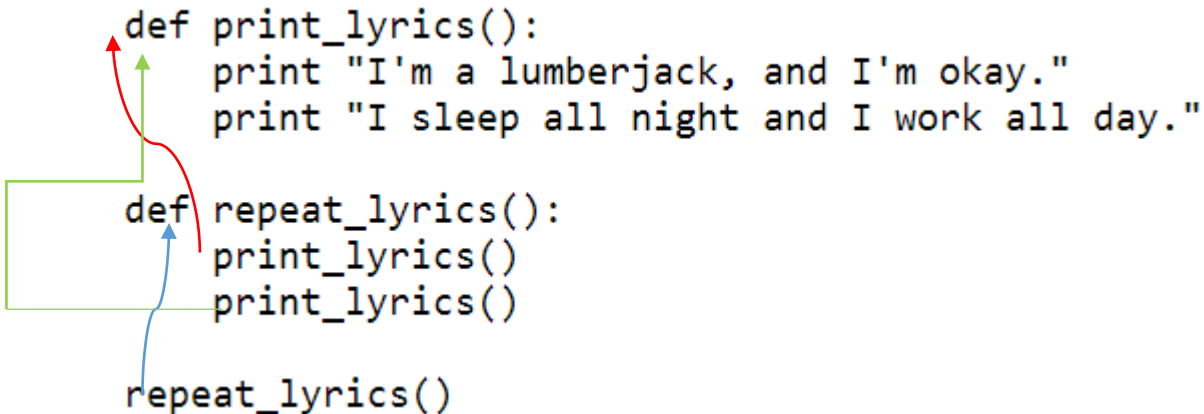
- ❖ Các câu lệnh bên trong hàm không được thực hiện cho đến khi hàm được gọi
- ❖ Định nghĩa hàm trước khi gọi hàm

- ❖ Bắt đầu với câu lệnh thứ nhất của chương trình. Các câu lệnh được thực hiện lần lượt từ trên xuống.
- ❖ Các định nghĩa hàm không làm thay đổi luồng thực hiện chương trình
- ❖ Mỗi lần gọi hàm là một lần rẽ ngoặt luồng thực hiện. Thay vì chuyển sang câu lệnh kế tiếp, luồng sẽ nhảy tới phần thân của hàm

```
def print_lyrics():
    print "I'm a lumberjack, and I'm okay."
    print "I sleep all night and I work all day."

def repeat_lyrics():
    print_lyrics()
    print_lyrics()

repeat_lyrics()
```



- ❖ Hàm cần 1 đối số: *math.sin*
- ❖ Hàm cần 2 đối số: *math.pow*
- ❖ Bên trong hàm, các đối số được gán cho các biến được gọi là tham

```
def print_twice(bruce):
    print bruce
    print bruce
>>> print_twice('Spam')
Spam
Spam
>>> print_twice(17)
17
17
>>> print_twice(math.pi)
3.14159265359
3.14159265359
```

Bạn cũng có thể dùng một biến cho một đối số

```
>>> michael = 'Eric, the half a bee.'
>>> print_twice(michael)
Eric, the half a bee.
Eric, the half a bee.
```

3.7 Các biến và tham số có tính chất cục bộ

❖ Khi tạo ra một biến ở trong hàm, nó mang tính cục bộ,

```
def cat_twice(part1, part2):  
    cat = part1 + part2  
    print_twice(cat)
```

```
>>> line1 = 'Bing tiddle '  
>>> line2 = 'tiddle bang.'  
>>> cat_twice(line1, line2)  
Bing tiddle tiddle bang.  
Bing tiddle tiddle bang.
```

```
>>> print cat  
NameError: name 'cat' is not defined
```

3.8 Hàm có và không trả lại kết quả

❖ Hàm trả lại kết quả: `math.sqrt()`, `math.cos()`

```
x = math.cos(radians)
golden = (math.sqrt(5) + 1) / 2
```

```
>>> math.sqrt(5)
2.2360679774997898
```

Trong script, nếu bạn gọi hàm trả kết quả mà ko được ghi lại thì kết quả sẽ mất vĩnh viễn

❖ Hàm không trả lại kết quả: `math.print_twice`

```
>>> result = print_twice('Bing')
Bing
Bing
>>> print result
None
```

```
>>> print type(None)
<type 'NoneType'>
```

3.9 Tại sao phải sử dụng hàm

- ❖ Đặt tên cho một nhóm các câu lệnh, từ đó làm cho chương trình dễ đọc và gỡ lỗi hơn.
- ❖ Các hàm có thể thu gọn một chương trình bằng cách loại bỏ những đoạn mã lệnh trùng lặp. Sau này, nếu bạn sửa đổi chương trình, thì chỉ cần thực hiện sửa ở một chỗ.
- ❖ Việc chia một chương trình dài thành những hàm cho phép ta gỡ lỗi từng phần một và sau đó kết hợp lại để được một chương trình tổng thể hoạt động được.
- ❖ Các hàm được thiết kế tốt sẽ hữu dụng với nhiều chương trình. Một khi bạn viết ra một hàm và gỡ lỗi xong xuôi, bạn có dùng lại nó.

3.10 Exercise

```
def do_twice(f, val):  
    f(val)  
    f(val)  
  
def print_twice(printVal):  
    print printVal  
    print `tran_manh_huy`  
  
def do_four(f, val):  
    do_twice(f, val)  
    do_twice(f, val)  
  
text = 'spam'  
  
do_four(print_twice, text)
```

3. Exercise

```
+ - - - - + - - - - +  
|           |           |  
|      *    |      *    |  
|           |           |  
+ - - - - + - - - - +  
|           |           |  
|      *    |      *    |  
|           |           |  
+ - - - - + - - - - +  
|           |           |  
|      *    |      *    |  
|           |           |  
+ - - - - + - - - - +
```

Hỏi đáp