LẬP TRÌNH PYTHON

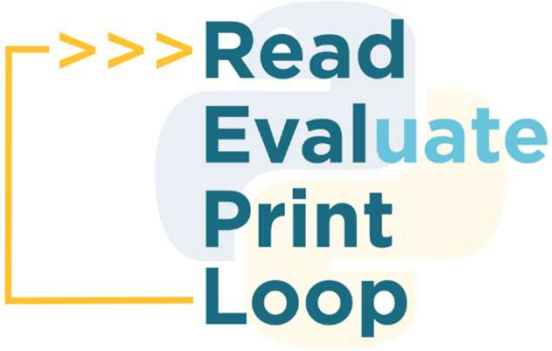
KIỂU DỮ LIỆU, TOÁN TỬ, CẤU TRÚC ĐIỀU KHIỂN

hungdn@ptit.edu.vn

Tương tác với Python

REPL

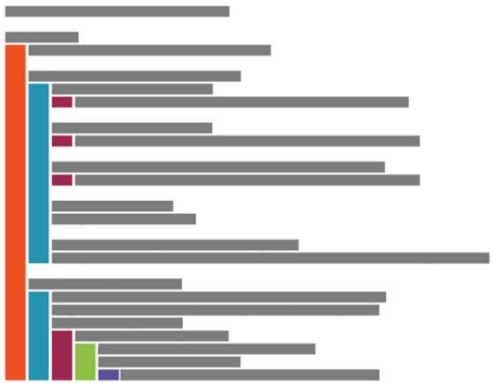
1. Read the user input
2. Evaluate your code
3. Print any results
4. Loop back to step 1



Khoảng trắng và thụt dòng

Quy tắc:

1. 4 khoảng trắng
2. Không sử dụng lẫn khoảng trắng với tab
3. Có thể sử dụng số lượng thụt dòng khác nhau tại các thời điểm tuy nhiên nhất quán trên các dòng liên tiếp
4. Có thể ngoại lệ nếu để cải thiện khả năng đọc của mã



The Zen Of Python

```
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

Thư viện chuẩn của python

- Từ khóa import
  - `import module_name`
  - `import from`
  - `from module import name`
  - `from module import name as name 2`
- Sử dụng từ khóa
  - dir
  - help

1. Kiểu dữ liệu (Scalar Type)

int

- **int : Số nguyên không giới hạn**

```
>>> 10
10
>>> 0b10
2
>>> 0o10
8
>>> 0x10
16
>>> int(3.5)
3
>>> int("496")
496
>>> int("10000", 3)
81
```

float

- **float : theo IEEE-754 dấu phẩy động độ chính xác kép (double) [53 bit cho độ chính xác – tương ứng 15-16 chữ số có nghĩa trong hệ thập phân]**

```
>>> 3.125
3.125
>>> 3e8
300000000.0
>>> 1.616e-35
1.616e-35
>>> float(7)
7.0
>>> float("1.618")
1.618
>>> float("nan")
nan
>>> float("inf")
inf
>>> 3.0 + 1
4.0
```

None

None

- **Giá trị rỗng (Null)**
- **Thường được sử dụng để mô tả sự vắng mặt của giá trị**

```
>>> None
>>>
>>> a = None
>>> a is None
True
```

bool

bool

- **Kiểu dữ liệu logic**
- **Hai giá trị: True và False**

```
>>> bool([])
False
>>> bool([1, 5, 9])
True
>>> bool("")
False
>>> bool("Spam")
True
```

```
>>> True
True
>>> False
False
>>> bool(0)
False
>>> bool(42)
True
>>> bool(-1)
True
>>> bool(0.0)
False
>>> bool(-1.117)
True
>>> bool(float("NaN"))
True
```

2. Toán tử (Operator)

Toán tử (1)

- Toán học
  - / Phép chia (trả về kiểu float)
  - // Phép chia (trả về phần nguyên)
  - ** Hàm mũ
- Logic
  - and
  - or
  - Not
- Không có toán tử tăng (++) và giảm (--)

Toán tử (2)

- **Identity operators**
  - is trả về True nếu các toán hạng tham chiếu cùng đối tượng
  - is not trả về True nếu các toán hạng tham chiếu khác đối tượng
- **Identical – giống hệt nhau**

```
x1 = 5
y1 = 5
x2 = 'Python'
y2 = 'Python'
x3 = [1,2,3]
y3 = [1,2,3]

# Output: False
print(x1 is not y1)

# Output: True
print(x2 is y2)

# Output: False
print(x3 is y3)
```

Toán tử (3)

- **Membership operators**
  - in True nếu giá trị/biến tìm thấy
  - not in True nếu giá trị/biến không tìm thấy

```
x = 'Python PTIT'
y = {1:'a',2:'b'}

# Output: True
print('P' in x)

# Output: True
print('PTIT' not in x)

# Output: True
print(1 in y)

# Output: False
print('a' in y)

# Output: False
list = [10, 20, 30, 40, 50]
print(24 in list)
```

Toán tử (4)

- **Membership operators**
  - in True nếu giá trị/biến tìm thấy
  - not in True nếu giá trị/biến không tìm thấy

```
x = 'Python PTIT'
y = {1:'a',2:'b'}
list = [10, 20, 30, 40, 50]

# Output: True
print('P' in x)

# Output: True
print('PTIT' not in x)

# Output: True
print(1 in y)

# Output: False
print('a' in y)

# Output: False
print(24 in list)
```

Toán tử (4)

- **Ternary operators**

```
[on_true] if [expression/condition] else [on_false]
```

```
a, b = 10, 20

# Ternary Operator
min = a if a < b else b
print(min)

# Or
print(a) if (a>b) else print(b)
```

Chú thích (comment)

- Chú thích là một tính năng cực kỳ hữu ích trong hầu hết các ngôn ngữ lập trình. Mọi thứ ta đã viết trong các chương trình cho đến nay đều là mã Python.
- Khi các chương trình trở nên dài hơn và phức tạp hơn, ta nên thêm ghi chú trong các chương trình mô tả cách tiếp cận tổng thể đối với vấn đề bài toán đang giải quyết

3. Cấu trúc điều khiển (Control Flow)

Câu lệnh rẽ nhánh/điều kiện

- **Nhánh được thực thi dựa trên giá trị của một biểu thức logic (True/False)**
- **Cú pháp**

```
if [expression]:
      [block]
```

```
if [expression]:
      [block]
elif [expression]:
      [block]
else:
      [block]
```

```
# if-elif-else ladder

i = 20
if (i == 10):
      print("i is 10")
elif (i == 15):
      print("i is 15")
elif (i == 20):
      print("i is 20")
else:
      print("i is not present")
```

Vòng lặp while

- **Thực thi dựa trên giá trị của biểu thức logic (True/False)**
- **Cú pháp**

```
while [expression]:
      [block]
```

```
# while loop

current_number = 1
while current_number <= 5:
    print(current_number)
    current_number += 1
```

Từ khóa break, continue

- **C, C++, C# và Java có vòng lặp do – while**

```
int choice = 0;
do{
  scanf("%d",&choice);
}while(choice %5 !=0);
```

- **Python sử dụng while và break**

```
>>> while True:
...     choice = input()
...     if int(choice) % 5 == 0
...         break
```

- **Từ khóa continue**

```
>>> i = 0
>>> while i < 10:
...     i += 1
...     if i % 2 == 0:
...         Continue
...     print(current_number)
```

Kết luận

- **REPL**
- **Từ khóa hữu ích import, dir, help**
- **Kiểu dữ liệu int float None và bool**
- **Toán tử**
- **Cấu trúc điều khiển if và while**

DEMO

- **import**
  - Import from
  - From module import name
  - From module import name as name 2
- **Dir, help**
- **math.factorial()**
- **math.gcd()**
- **input()**
- **print() one line**
- **Import operator**
- **So sánh 02 số**
- **Vòng lặp**
- **range()**
- **Pythonic way**

Bài tập

- **Tìm số lớn nhất/nhỏ nhất trong 03 số**
- **Các bài vẽ hình (vòng lặp) / Số**
- *
- **
- ***
- ****

Q & A