

# Các cấu trúc cơ bản trong Python

---

# Nội dung

---

1. **Toán tử trong python**
2. **Nhập, xuất và tính toán**
3. **Cấu trúc rẽ nhánh**
4. **Q & A**

# 1.1. Các loại toán tử

---

❖ Python hỗ trợ các loại toán tử:

- Toán tử số học
- Toán tử quan hệ (còn gọi là toán tử so sánh)
- Toán tử gán
- Toán tử logic
- Toán tử thao tác bit

## 1.2. Toán tử số học

---

Toán tử	Miêu tả
//	Thực hiện phép chia lấy phần nguyên
+	Phép cộng
-	Phép trừ
*	Phép nhân
/	Phép chia
%	Phép chia lấy phần dư
**	Phép lấy số mũ (ví dụ $2^{**}3$ cho kết quả là 8)

---

*# Ví dụ:*

```
_a = 14
```

```
_b = 2
```

```
_c = 4
```

```
print(_a + _b) #16
```

```
print(_a - _b) #12
```

```
print(_a * _b) #28
```

```
print(_a // _c) #3
```

```
print(_a % _c) #2
```

```
print(_a / _c) #3.5
```

```
print(_b ** _c) #16
```

## 1.3. Toán tử quan hệ (so sánh)

---

❖ Toán tử so sánh trả về kết quả logic **True** (1) hoặc **False** (0). Dùng được cho cả number và string.

Toán tử	Miêu tả
<	Nhỏ hơn (Giá trị của toán hạng trái là nhỏ hơn giá trị của toán hạng phải → true).
>	Lớn hơn
<=	Nhỏ hơn hoặc bằng
>=	Lớn hơn hoặc bằng
==	Bằng
!=	Không bằng

---

```
# Ví dụ  
_a = 14  
_b = 2  
_c = 4  
_d = 2
```

```
print(_b < _a)           #True  
print(_b > _a)           #False  
print(_b <= _a)         #True  
print(_b >= _a)         #False  
print(_b == _d)         #True  
print(_b != _a)         #True
```

## 1.4. Toán tử gán

❖ Thực hiện phép tính của giá trị trái cho giá trị phải, và gán kết quả cho giá trị trái:

Toán tử	Miêu tả
=	Phép gán bằng
/=	Chia và gán
+=	Cộng và gán
-=	Trừ và gán
*=	Nhân và gán
%=	Chia lấy phần dư và gán
**=	Lấy số mũ và gán
//=	Thực hiện phép chia // và gán



```
#
_C = 10
print(_C)
_C += 5
print(_C)
_C -= 5
print(_C)
_C *= 2
print(_C)
_C /= 2
print(_C)
_C %= 3
print(_C)
#
_D = 5
print(_D)
_D **= 2
print(_D)
_D //= 2
print(_D)
```

## 1.5. Toán tử logic

---

❖ Toán tử logic cũng trả về kết quả logic **True** hoặc **False**.

Toán tử	Miêu tả
and	Phép và, nếu cả hai điều kiện là true thì kết quả sẽ là true
or	Phép hoặc, nếu một trong hai điều kiện là true thì kết quả là true
not	Phép phủ định. Được sử dụng để đảo ngược trạng thái logic của toán hạng

---

```
# Ví dụ
_a = 5 > 4 and 3 > 2
print(_a)          #True and true = True
_b = 5 > 4 or 3 < 2
print(_b)          #True or False = True
_c = not (5 > 4)
print(_c)          #Not(True) = False
```

## 1.6. Toán tử thao tác bit

---

Toán tử	Miêu tả
&	Phép <b>cộng (AND)</b> logic của bit.
	Phép <b>hoặc (OR)</b> logic của bit.
^	Phép <b>XOR</b> của bit.
~	Phép <b>phủ định (NOT)</b> logic của bit.
<<	Toán tử dịch trái nhị phân.
>>	Toán tử dịch phải nhị phân.

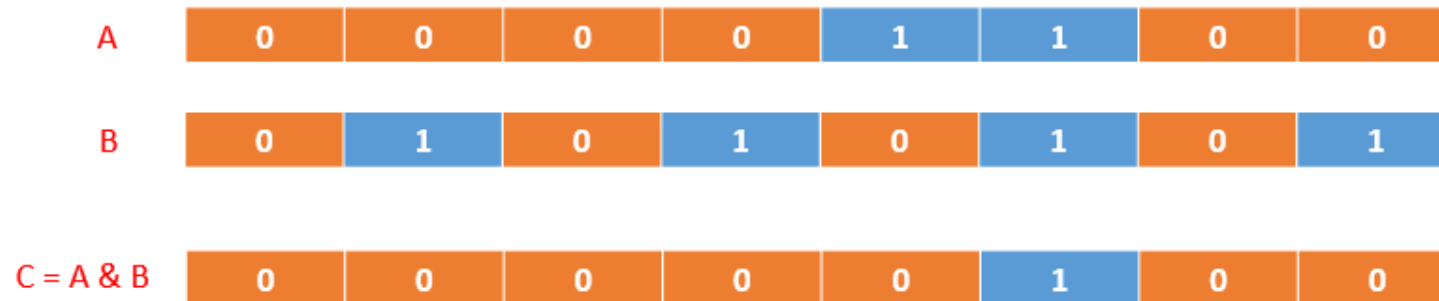
# AND

❖ Kí hiệu: &

❖  $A = 12$  (00001100)

❖  $B = 85$  (01010101)

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1



# OR

❖ Kí hiệu: |

A	B	A   B
0	0	0
0	1	1
1	0	1
1	1	1

A	0	0	0	0	1	1	0	0
B	0	1	0	1	0	1	0	1
C = A   B	0	1	0	1	1	1	0	1

# NOT

❖ Kí hiệu:  $\sim$

A	$\sim A$
0	1
1	0

A	0	0	0	0	1	1	0	0
B = $\sim A$	1	1	1	1	0	0	1	1

# XOR

❖ Kí hiệu:  $\wedge$

A	B	$A \wedge B$
0	0	0
0	1	1
1	0	1
1	1	0

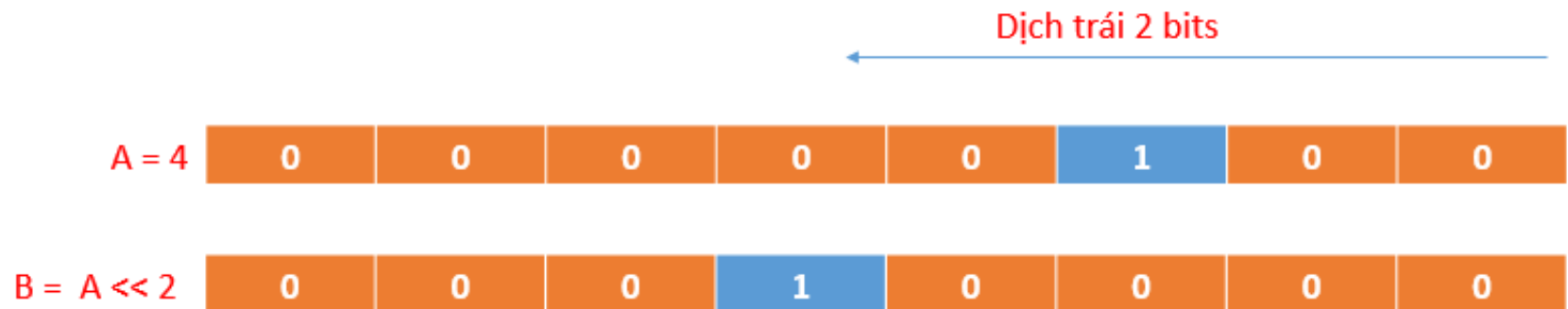
A	0	0	0	0	1	1	0	0
B	0	1	0	1	0	1	0	1
$C = A \wedge B$	0	1	0	1	1	0	0	1



# Phép dịch trái <<

❖ Kí hiệu: <<

❖ Phép dịch trái  $n$  bit tương đương với phép nhân cho 2 mũ  $n$ .

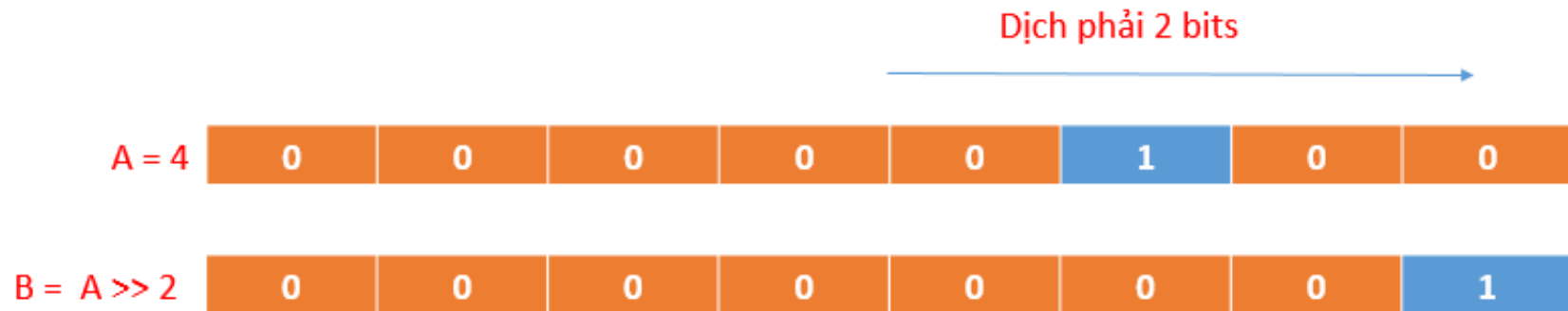


# Phép dịch phải >>

---

❖ Kí hiệu: >>

❖ Phép dịch phải  $n$  bit tương đương với phép chia cho  $2$  mũ  $n$ .



## 1.7. Luyện tập (1)

---

❖ **Bài 1:** Cho các giá trị  $\_a = 16$ ,  $\_b = 3$ ,  $\_c = 5$ . Viết một lớp python để thực hiện lần lượt các phép toán sau đối với 3 biến  $\_a$ ,  $\_b$ ,  $\_c$  và in ra kết quả:

- Các phép toán số học: *cộng; trừ; nhân; chia; lũy thừa...*
- Toán tử quan hệ (còn gọi là toán tử so sánh): *lớn hơn; nhỏ hơn; bằng; khác...*
- Toán tử gán: *cộng gán; chia gán; nhân gán...*
- Toán tử logic: *and; or; not*
- Toán tử thao tác bit: *and; or; not; xor; dịch bit trái 3 vị trí; dịch bit phải 2 vị trí*

## 1.7. Luyện tập (2)

---

❖ **Bài 2:** Cho các giá trị `_s1 = 'a'`, `_s2 = 'b'`, `_s3 = 'c'`. Viết một lớp python để thực hiện lần lượt các phép toán sau đối với 3 biến `_s1`, `_s2`, `_s3` và in ra kết quả:

- ❑ Toán tử quan hệ (còn gọi là toán tử so sánh): *lớn hơn; nhỏ hơn; bằng; khác...*
- ❑ Toán tử logic: *and; or; not*

- 
1. Toán tử trong python
  2. **Nhập, xuất và tính toán**
  3. Cấu trúc rẽ nhánh
  4. Q & A

## 2.1. Nhập dữ liệu từ bàn phím

---

❖ Từ khóa: `input`

❑ Cú pháp cơ bản:

```
_i = input("Nhập giá trị đầu vào: ")
```

❑ Cú pháp ép kiểu:

```
_i = float(input("Nhập giá trị đầu vào: "))  
_j = int(input("Nhập giá trị đầu vào: "))
```

## 2.2. Xuất dữ liệu ra console

---

❖ Từ khóa: `print`

☐ Cú pháp cơ bản:

```
print('Giá trị số thực:', _i)
print('Giá trị số nguyên: ', _j)
print('Giá trị số thực:', _i, '; Giá trị số nguyên: ', _j)
```

## 2.3. Luyện tập

---

❖ **Bài 1:** Nhập vào từ bàn phím hai số nguyên. Tính tổng và in ra tổng của hai số nguyên đó.

❖ **Bài 2:** Nhập vào từ bàn phím chuỗi ký tự, in ra chuỗi ký tự đó.

❖ **Bài 3:** Nhập vào từ bàn phím ba số nguyên. Tính toán và in ra màn hình:

a) Tổng và tích của ba số đó

b) Hiệu của 2 số bất kỳ trong 3 số đó

c) Phép chia lấy phần nguyên, phần dư và kết quả chính xác của 2 số bất kỳ trong 3 số đó

❖ **Bài 4:** Nhập vào từ bàn phím ba chuỗi ký tự. In ra màn hình một chuỗi ký tự được ghép từ ba chuỗi nhập vào. Ví dụ: Nhập vào 'IT', 'PL', 'US' thì kết quả in ra là ITPLUS.

❖ **Bài 5:** Tính toán và in ra Chu vi, diện tích của hình tròn. Với giá trị bán kính được nhập từ bàn phím. Công thức tính chu vi:  $CV = 2 * \_R * \_pi$ ;  $DT = \_pi * \_R * \_R$  (trong đó:  $\_pi$  là số  $pi = 3.14$ ,  $\_R$  là bán kính đường tròn)



- 
1. Toán tử trong python
  2. Nhập, xuất và tính toán
  3. **Cấu trúc rẽ nhánh**
  4. Q & A

## 3.1. Cấu trúc điều khiển luồng dữ liệu

---

❖ Từ khóa: **if – else** hoặc **if – elif – else**

❖ Cú pháp:

```
if bieu_thuc_1 :  
    # Làm gì đó
```

```
elif bieu_thuc_2:  
    # Làm gì đó
```

```
elif bieu_thuc_N:  
    # Làm gì đó
```

```
else :  
    # Làm gì đó
```

---

❖ Ví dụ:

```
option = 2

if option == 1:
    print("Hello")
else :
    print("Bye!")
```

## 3.2. Các toán tử so sánh thông dụng

---

Toán tử	Ý nghĩa	Ví dụ
>	Lớn hơn	$5 > 4$ là đúng (true)
<	Nhỏ hơn	$4 < 5$ là đúng (true)
>=	Lớn hơn hoặc bằng	$4 >= 4$ là đúng (true)
<=	Nhỏ hơn hoặc bằng	$3 <= 4$ là đúng (true)
==	Bằng nhau	$1 == 1$ là đúng (true)
!=	Không bằng nhau	$1 != 2$ là đúng (true)
and	Và	$a > 4$ and $a < 10$
or	Hoặc	$a == 1$ or $a == 4$

```
print("Please enter your age: \n")  
# Khai báo một biến inputStr, lưu trữ dòng text người dùng nhập vào từ bàn phím.  
inputStr = input()  
# Hàm int(..) chuyển một chuỗi thành 1 số tự nhiên  
age = int(inputStr)
```

---

```
# In ra tuổi của bạn  
print("Your age: ", age)
```

```
# Kiểm tra nếu age nhỏ hơn 80 thì ...  
if (age < 80) :  
    print("You are pretty young")
```

```
# Ngược lại nếu tuổi nằm trong khoảng 80, 100 thì  
elif (age >= 80 and age <= 100) :  
    print("You are old")
```

```
# Ngược lại (Các trường hợp còn lại)  
else :  
    print("You are verry old")
```

- 
- ❖ Trong 1 luồng điều khiển dữ liệu, chỉ duy nhất 1 khối lệnh bên trong **if** hoặc **elif** hoặc **else** được thực thi.
  - ❖ Khối code sẽ được thực thi nếu biểu thức điều kiện trong lệnh **if** hoặc **elif** được ước lượng **khác 0** hoặc một giá trị **true**.
  - ❖ Ngược lại, khối code trong lệnh **else** sẽ được thực thi nếu tất cả các biểu thức điều kiện trong lệnh **if** và **elif** được ước lượng là **0** hoặc một giá trị **false**.

## ❖ Ví dụ

```
_var1 = -100
_var2 = 100
if _var1:
    print("1.1 - True")
    print(_var1)
elif True:
    print("1.2 - True")
    print(_var2)
else:
    print("1.3 - Nhan mot gia tri False")
    print(_var1 + _var2)

_var3 = 0
if _var3:
    print("2.1 - Nhan mot gia tri True")
    print(_var3)
else:
    print("2.2 - Nhan mot gia tri False")
    print(_var3)
```

## 3.3. Luyện tập

---

❖ **Bài 1:** Viết chương trình nhập vào một số nguyên dương và in "Đây là một số chẵn" nếu nó chẵn và in "Đây là một số lẻ" nếu là số lẻ.

❖ **Bài 2:** Nhập vào 3 số nguyên dương `_a`, `_b`, `_c` và in "Độ dài ba cạnh tam giác" nếu nó thỏa mãn và in "Đây không phải độ dài ba cạnh tam giác" nếu không thỏa mãn.

❖ **Bài 3:** Viết chương trình nhập vào năm sinh, in ra tuổi. Ví dụ nhập 1990 in ra: Năm sinh 1990, vậy bạn 28 tuổi. Câu lệnh để lấy ra năm hiện tại:

```
import time

x = time.localtime()
year = x[0]
```



## 3.4. Lồng các lệnh If

---

- ❖ Đôi khi có vài tình huống muốn kiểm tra thêm một điều kiện khác sau khi một điều kiện đã được ước lượng là **true**. Trong tình huống như vậy, có thể sử dụng các lệnh if lồng nhau trong Python.
- ❖ Trong cấu trúc các lệnh if lồng nhau, có thể có cấu trúc if...elif...else bên trong cấu trúc if...elif...else khác.

---

## ❖ Cú pháp:

```
if bieu_thuc1:
    cac_lenh
    if bieu_thuc2:
        cac_lenh
    elif bieu_thuc3:
        cac_lenh
    else:
        cac_lenh
elif bieu_thuc4:
    cac_lenh
else:
    cac_lenh
```

```
_number = 10
if _number < 15:
    print ("Nhỏ hơn 15.")
    if _number == 15:
        print ("Giá trị là 15.")
    elif _number == 10:
        print ("Giá trị là 10.")
    elif _number == 5:
        print ("Giá trị là 5.")
elif _number < 5:
    print ("Nhỏ hơn 5.")
else:
    print ("Không tìm thấy giá trị thích hợp")

print ("Finish!")
```

## 3.5. Luyện tập

---

❖ **Bài 4:** Viết chương trình nhập một số nguyên dương và kiểm tra xem số đó có chia hết cho 2 hoặc cho 3 hoặc cả hai hay không?

❖ **Bài 5:** Viết chương trình giải phương trình bậc 2:  $a*x*x + b*x + c = 0$ . Trong đó 3 tham số  $a, b, c$  được nhập từ bàn phím. Sử dụng thư viện `math` và hàm tính căn `sqrt()`. Ví dụ tính căn của 4:

```
import math

_sqrt_4 = math.sqrt(4)
```

