

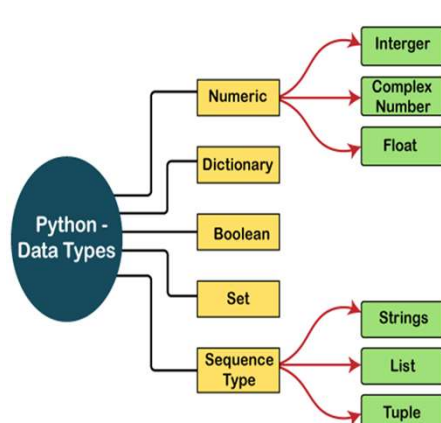
CƠ BẢN TRONG PYTHON

- 01** Kiểu dữ liệu cơ bản
- 02** Cách ghi chú
- 03** Toán tử
- 04** Nhập/Xuất
- 05** Xử lý lỗi

1

1. Kiểu dữ liệu cơ bản

Kiểu dữ liệu cơ bản	Mô tả	Ví dụ
int	Số nguyên	a=100 b=-100
float	Số thực	Pi=3.14
complex	Số phức	c1= 1+2i c2=Complex(1, 2) c1.real, c2.imag
str	Chuỗi	s='Khoa CNTT'
bool	Kiểu logic: True, False	kt=True



2

2

1. Kiểu dữ liệu cơ bản

- **Khai báo biến:** Ta không cần khai báo kiểu cho biến. Việc dùng kiểu dữ liệu nào do Python tự xác định! Chỉ cần gán một giá trị trong lúc khai báo biến, điều đó sẽ giúp Python nhận biết được biến đó sẽ lưu trữ kiểu dữ liệu gì.
`tên_biến = giá_trị`
- **Xóa biến:** Xóa biến để giải phóng vùng nhớ. Sau khi xóa 1 biến ta không thể truy cập đến biến đó
`del tên_biến`

```

hoten='Lan Huong'
trongLuong = 50.5
print(hoten)
del hoten
print(hoten)

```

Lan Huong
 Traceback (most recent call last):
 File "D:\Subjects\Python_Program\Example\Xuat.py", line 5, in <module>
 print(hoten)
 NameError: name 'hoten' is not defined. Did you mean: 'open'?

3

3

1. Kiểu dữ liệu cơ bản

- **Có thể xem kiểu dữ liệu của biến bằng cách sử dụng type()**

```

x=5
print(type(x))
x='CNTT'
print(type(x))
x=True
print(type(x))
x=5.5
print(type(x))
x=complex(113,114)
print(type(x))

```

<class 'int'>
 <class 'str'>
 <class 'bool'>
 <class 'float'>
 <class 'complex'>

- **Xem vùng lưu trữ giá trị của các biến int, float**

```

import sys
print("Thông tin chi tiết của int:")
print(sys.int_info)
print("Thông tin chi tiết của float:")
print(sys.float_info)

```

Thông tin chi tiết của int:
 sys.int_info(bits_per_digit=15, sizeof_digit=2)
 Thông tin chi tiết của float:
 sys.float_info(max=1.7976931348623157e+308, max_exp=1024, max_10_exp=308, min=2.2250738585072014e-308, min_exp=-1021, min_10_exp=-307, dig=15, mant_dig=53, epsilon=2.220446049250313e-16, radix=2, rounds=1)

4

4

2. Cách ghi chú

Python dùng từ khóa:

- #: cho phép ghi chú 1 dòng
- """ """ (3 cặp nháy đôi) hoặc ''' ''' (3 cặp nháy đơn): ghi chú nhiều dòng

Ghi chú 1 dòng →

Ghi chú nhiều dòng →

```
#Xuất chuỗi
print('Xin chào khoa CNTT')

'''
Nhập thông tin:
- Họ tên
- Mã số
- Điểm
'''
```

Xin chào khoa CNTT

5

3. Toán tử

Toán tử	Mô tả	Ví dụ
+	Cộng	12 + 4.9 => kết quả 16.9
-	Trừ	3.98 - 4 => kết quả -0.02
*	Nhân	2 * 3.4 => kết quả 6.8
/	Chia	9 / 2 => kết quả 4.5
//	Chia lấy phần nguyên	9 // 2 => kết quả 4
%	Chia lấy phần dư	9%2 => kết quả 1
**	Lũy thừa	3**4=>kết quả 81

6

3. Toán tử			
Toán tử	Mô tả	Ví dụ	Tương đương với
=	Phép gán giá trị bên phải cho biến bên trái dấu bằng	x=5	
+=	Cộng và gán	x=2 x+=5 ==>x=7	x=x+5
-=	Trừ và gán	x=2 x-=5 ==>x=-3	x=x-5
=	Nhân và gán	x=2 x=5 ==>x=10	x=x*5

7

7

3. Toán tử			
Toán tử	Mô tả	Ví dụ	Tương đương với
/=	Chia và gán	x=7 x/=5 ==>x=1.4	x=x/5
//=	Chia và gán (lấy nguyên)	x=7 x//=5 ==>x=1	x=x//5
%=	Chia lấy dư	x=7 x%=5 ==>x=2	x=x%5
=	Lấy lũy thừa và gán	x=2 x=3 ==>x là 2 mũ 3 =8	x=x**3

8

8

3. Toán tử		
Toán tử	Mô tả	Ví dụ
==	So sánh bằng	5 == 5 => kết quả True
!=	So sánh không bằng	5 != 5 => kết quả False
<	So sánh nhỏ hơn	5 < 5 => kết quả False
<=	So sánh nhỏ hơn hoặc bằng	5 <= 5 => kết quả True
>	So sánh lớn hơn	5 > 5.5 => kết quả False
>=	So sánh lớn hơn hoặc bằng	113 >= 5 => kết quả True
is	Trả về true nếu các biến ở hai bên toán tử cùng trỏ tới một đối tượng(hoặc cùng giá trị), nếu không là false	x=5 y=5 print(x is y) =>kết quả là True
is not	Trả về false nếu các biến ở hai bên toán tử cùng trỏ tới một đối tượng(hoặc cùng giá trị), nếu không là true	x=5 y=5 print(x is not y) =>kết quả là False

9

3. Toán tử		
Toán tử	Mô tả	Ví dụ
and	Toán tử Và: Nếu cả hai điều kiện là True thì kết quả sẽ là True	x=2016 print(x%4==0 and x%100!=0) =>True
or	Toán tử Hoặc: Chỉ cần một điều kiện True thì nó True, tất cả điều kiện False thì nó False	x=2016 print((x%4==0 and x%100!=0) or x%400==0) =>True
not	Toán tử Phủ định. Thông thường nó được dùng để đảo ngược trạng thái logic của toán hạng	x=4 if (not x>=5): print("Ngắm gà khóa thân và nài chuỗi") else: print("Đậu")

10

10

3. Toán tử		
■ Độ ưu tiên của toán tử		
Thứ tự ưu tiên	Toán tử	Miêu tả
1	()	Đóng mở ngoặc
2	**	Toán tử mũ
3	* / % //	Phép nhân, chia, lấy phần dư và phép chia lấy phần nguyên
4	+ -	Toán tử Cộng, Trừ
5	<= < > >=	Các toán tử so sánh
6	<> == !=	Các toán tử so sánh
7	= %= /= //=- += *= **=	Các toán tử gán
8	is, is not	Các toán tử so sánh
9	not, or, and	Các toán tử Logic

11

<pre> a=10 b=2 c=not(2**3 + a//3 + b%5 <= 5 and 3=4) </pre>		
--	--	--

12

4. Nhập/Xuất

- Xuất: sử dụng hàm `print()` `print(10, end='\t')`

Ký tự đặc biệt (Escape Sequences)	Mô tả
<code>\n</code>	Xuống dòng mới
<code>\r</code>	Chuyển con trỏ đến đầu dòng hiện hành
<code>\t</code>	Chuyển con trỏ đến vị trí dừng Tab kế tiếp (ký tự Tab)
<code>\\</code>	In dấu <code>\</code>
<code>\'</code>	In dấu nháy đơn (<code>'</code>)
<code>\''</code>	In dấu nháy kép (<code>"</code>)

13

4. Nhập/Xuất

- Xuất lập dữ liệu ra màn hình:
`print('*', 15)`
- Dùng hàm `format`
`print('{0}, {1}'.format(7, 10**7))`
- Mặc định hàm `print` căn trái, muốn căn phải sử dụng:
`print('-', 15)`
`print('{0:>2} {1:>11}'.format('STT', 'Giá trị'))`
`print('-', 15)`

```

STT  Giá trị
-----
100
3.141592653589793
a=100, PI=3.14

```

```

10000000
'7 10000000'

```

```

'y'
'a{0}b{1}c{0}d'.format('x', 'y')
'x'

```

```

a=100
PI=3.141592653589793
print(a)
print(PI)
print('a=%s, PI=%.2f'%(a, PI))

```

```

100
3.141592653589793
a=100, PI=3.14

```

14

4. Nhập/Xuất

- Nhập liệu từ bàn phím sử dụng hàm:
 - `input()`
 - `input(str)`: nhấn tiêu đề str
- Kiểu dữ liệu nhập: chuỗi
- Nếu muốn kiểu dữ liệu khác chuỗi ta sử dụng ép kiểu.

```

print('Nhap du lieu:')
x=input()
print('Ban vua nhap \'{0}\'' co kieu
{1}'.format(x, type(x)))

print('Nhap thong tin ca nhan')
hoten=input('Nhap ho ten: ')
namsinh=int(input('Nhap nam sinh:'))
dtb=float(input('Diem trung binh:'))
print('Ban ten \'{0}\', sinh nam {1}, diem
tb {2}'.format(hoten, namsinh, dtb))
```

Nhap du lieu:
10
Ban vua nhap '10' co kieu <class 'str'>

Nhap thong tin ca nhan
Nhap ho ten: Nguyen Lan Huong
Nhap nam sinh:2000
Diem trung binh:8
Ban ten 'Nguyen Lan Huong ', sinh nam 2000,
diem tb 8.0

15

15

5. Xử lý lỗi

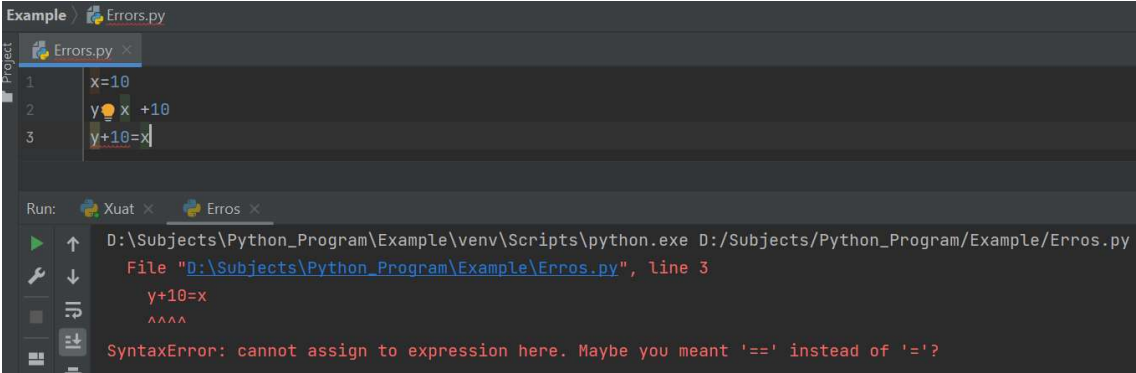
- Các loại lỗi:
 - Lỗi cú pháp (Syntax Errors)
 - Lỗi thực thi (Run-time Exceptions)
 - Lỗi nghiệp vụ (Logic Errors)
- Bắt lỗi

16

16

5. Xử lý lỗi

- Lỗi cú pháp (Syntax Errors)



```

1 x=10
2 y = x + 10
3 y+10=x

```

Run: D:\Subjects\Python_Program\Example\venv\Scripts\python.exe D:/Subjects/Python_Program/Example/Errors.py

File "D:\Subjects\Python_Program\Example\Errors.py", line 3

```

y+10=x
      ^
SyntaxError: cannot assign to expression here. Maybe you meant '==' instead of '='?

```

17

17

5. Xử lý lỗi

- Lỗi thực thi (Run-time Exceptions):
 - Thường lỗi này thông dịch hoàn tất, nhưng trong quá trình thực thi lại phát sinh ra lỗi.
 - Các lỗi thường gặp như chia cho 0, ép kiểu sai, đọc file lỗi, kết nối mạng lỗi....



```

1 sobichia=int(input('Nhập số bị chia:'))
2 sochia=int(input('Nhập số chia:'))
3 thuong=sobichia/sochia
4 print('{0}/{1}={2}'.format(sobichia,sochia, thuong))

```

Run: D:\Subjects\Python_Program\Example\venv\Scripts\python.exe D:\Subjects\Python_Program\Example\Errors.py

Nhập số bị chia: 10

Nhập số chia: 0

Traceback (most recent call last):

File "D:\Subjects\Python_Program\Example\Errors.py", line 3, in <module>

```

    thuong=sobichia/sochia
ZeroDivisionError: division by zero

```

18

18

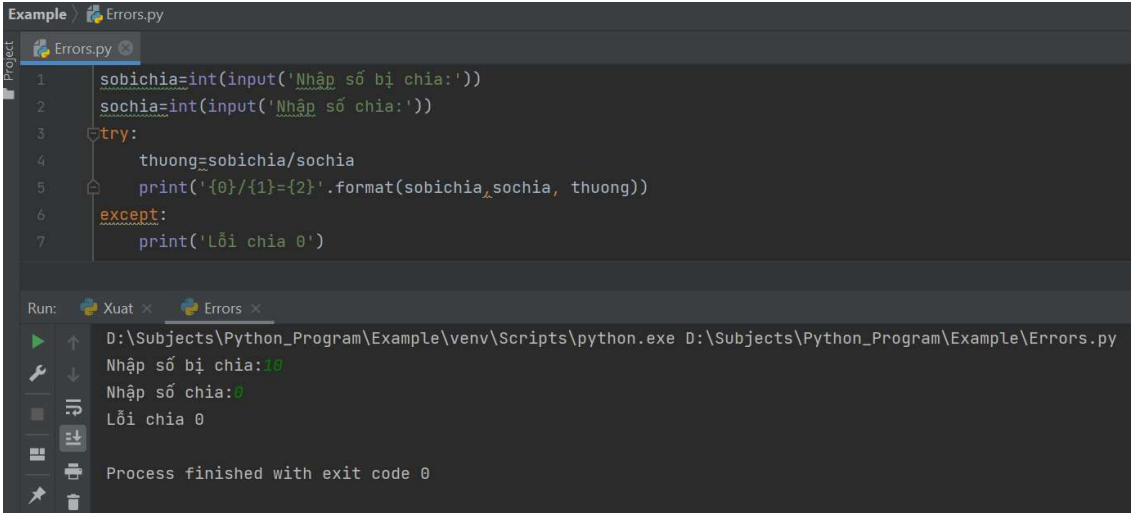
5. Xử lý lỗi

- Lỗi nghiệp vụ (Logic Errors)
 - Là lỗi nghiêm trọng sai yêu cầu nghiệp vụ của khách hàng
 - Khó phát hiện

19

5. Xử lý lỗi

- Bắt lỗi: sử dụng try ... except



```

Example > Errors.py
Errors.py
1 sobichia=int(input('Nhập số bị chia:'))
2 sochia=int(input('Nhập số chia:'))
3 try:
4     thuong=sobichia/sochia
5     print('{0}/{1}={2}'.format(sobichia,sochia, thuong))
6 except:
7     print('Lỗi chia 0')

Run: Xuat x Errors x
D:\Subjects\Python_Program\Example\venv\Scripts\python.exe D:\Subjects\Python_Program\Example\Errors.py
Nhập số bị chia:10
Nhập số chia:0
Lỗi chia 0
Process finished with exit code 0
  
```

20



THẢO LUẬN





21

21



Question?

22