



HCMUTE

TRƯỜNG ĐẠI HỌC

**SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

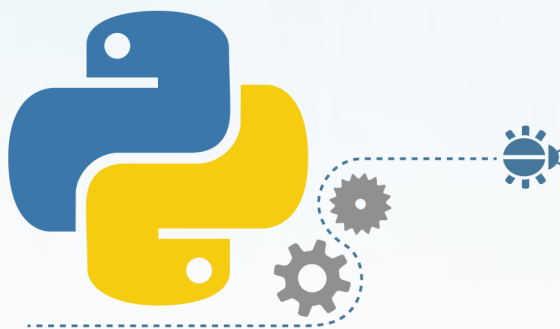
HCMC University of Technology and Education



KHOA CÔNG NGHỆ THÔNG TIN  
BỘ MÔN HỆ THỐNG THÔNG TIN

# NHẬP MÔN LẬP TRÌNH PYTHON (IPPA233277)

## CÁC KHÁI NIỆM CƠ BẢN



**GV. Trần Quang Khải**

1. Kiểu dữ liệu cơ bản & khai báo biến
2. Cách ghi chú lệnh
3. Toán tử
4. Nhập liệu từ bàn phím
5. Kiểu xuất dữ liệu
6. Các loại lỗi trong python



1. Hiểu và thực hiện được các kiểu dữ liệu cơ bản
2. Khai báo được biến, ghi chú
3. Thực hành được các toán tử
4. Nhập liệu được từ bàn phím
5. Xử lý được các lỗi trong Python



- Python hỗ trợ các kiểu dữ liệu số: **số nguyên (integer)**, **số thực (float)**, **số phức (complex)**
    - **Số nguyên (integer)** không chứa dấu chấm thập phân gồm cả số âm và dương và không giới hạn về độ dài  
Ví dụ: 218, -1991
    - **Số thực (float)** chứa dấu chấm thập phân (độ chính xác 15 chữ số phần thập phân) và có thể biểu diễn dạng e  
Ví dụ: 5.2, -7.3, 1.5e2 ( $\sim 1.5 \times 10^2$ )
    - **Số phức (complex)** biểu diễn dưới dạng  $x + yj$  (trong đó  $x$  là phần thực,  $y$  là phần ảo và bắt buộc để giá trị)  
Ví dụ: 45j, 3 + 2j, 1 + 3.2e25j, 6 + 1j, 8 - 0j
- Được biểu diễn `z = complex(3, 2)`. Khi cần xuất:
- ```
print("Phần thực: ", z.real) ➔ Phần thực: 3
print("Phần ảo: ", z.imag) ➔ Phần ảo: 2
```



- Python hỗ trợ biểu diễn các số ở hệ nhị phân (binary), bát phân (octal), thập lục phân (hexadecimal).

| Hệ số       | Tiền tố (prefix) | Giá trị | Ghi chú                               |
|-------------|------------------|---------|---------------------------------------|
| Binary      | '0b' hoặc '0B'   | 0B100   | Số dương 100 cơ số 2 (số 8 cơ số 10)  |
| Octal       | '0o' hoặc '0O'   | 0o100   | Số dương 100 cơ số 8 (số 64 cơ số 10) |
| Hexadecimal | '0x' hoặc '0X'   | -0X100  | Số âm 100 cơ số 16 (số -256 cơ số 10) |

- Kiểu chuỗi (str) để trong nháy đôi " ..." hoặc nháy đơn ' ... ' hoặc 3 nháy kép """ ... """, trong python không có kiểu dữ liệu ký tự (character)

Ví dụ: "HCMUTE", "FIT", """information technology, university of technology and education"""

- Kiểu luận lý (boolean) để lưu giá trị đúng (True, 1) hoặc sai (False, 0)

Ví dụ: t1 = True, t2 = False + 5



- Từ khóa (keyword) là những từ chỉ dành riêng cho python, không dùng để đặt tên. Trong python, ngoại trừ True, False, None viết hoa thì các từ khóa đều là dạng chữ thường. Để xem danh sách từ khóa:

```
>>> import keyword
```

```
>>> print(keyword.kwlist)
```

- Trong python không cần khai báo kiểu dữ liệu, khi thực hiện gán giá trị, python sẽ tự động nội suy kiểu dữ liệu. Ta có thể dùng hàm type để kiểm tra kiểu dữ liệu của biến.

```
x = 5
print(type(x))          <class 'int'>
x = 'fit-hcmute'
print(type(x))          <class 'str'>
x = True
print(type(x))          <class 'bool'>
```

```
x = 5.5
print(type(x))          <class 'float'>
x = complex(3, 2)
print(type(x))          <class 'complex'>
print(x.real, x.imag)    3 2
```





- None là một hằng số đặc biệt trong python, đại diện cho biến đó không có giá trị hoặc giá trị Null
- None là một đối tượng với kiểu dữ liệu riêng (NoneType), không phải 0, False, chuỗi rỗng, chỉ đơn giản là None và chứa giá trị như nhau dù gán cho các đối tượng khác nhau; có thể được trả về bởi một hàm không có lệnh trả về

```
>>> None == 0           # False           >>> x = None
>>> None == []          # False           >>> y = None
>>> None == False       # False           >>> x == y           # True
```

- Python dùng từ khóa del để xóa biến

```
x = "Obama"           # khởi tạo biến x với giá trị "Obama"
print(x)              Obama
del x                 # xóa biến x
print(x)              Traceback (most recent call last):
                      File "/XoaBien.py", line 4, in <module>
                        print(x)
                      NameError: name 'x' is not defined
```



- Kiểm tra vùng dữ liệu lưu trữ giá trị của các biến int, float bằng thư viện **sys**

```
import sys                                     # int
print("Thông tin chi tiết của int:")          sys.int_info(bits_per_digit=15, sizeof_digit=2)
print(sys.int_info)                           # float
print("Thông tin chi tiết của float:")        sys.float_info(max=1.7976931348623157e+308,
print(sys.float_info)                        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)
```

- **and, or, not** là các toán tử logic
- **as** được sử dụng để tạo bí danh khi đang nhập trong một module. `import numpy as np`
- **assert** được sử dụng cho mục đích gỡ lỗi



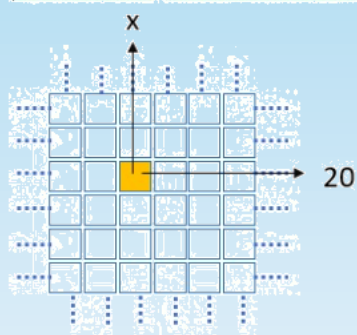
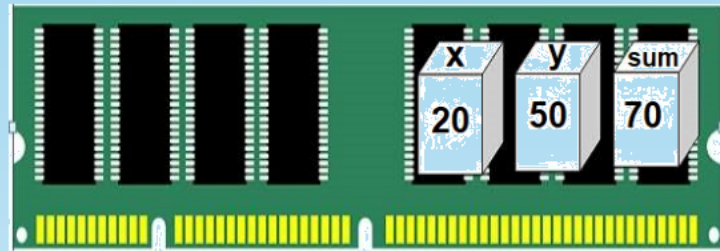


- Định danh là tên được đặt cho các thực thể như class, function, biến,... trong python.
- Quy tắc viết định dạng:
  - Là sự kết hợp giữa chữ thường (a – z), chữ hoa (A – Z), số (0 – 9) hoặc dấu gạch dưới (\_), không bắt đầu là số
  - Phải khác các từ khóa trong python, không hỗ trợ các ký tự đặc biệt (! @ \$ % ...)
  - Độ dài tùy ý, phân biệt chữ hoa – thường
  - Tên nên có ý nghĩa, có thể dùng dấu gạch dưới hoặc viết hoa đầu từ để phân biệt (check\_value, CheckValue)

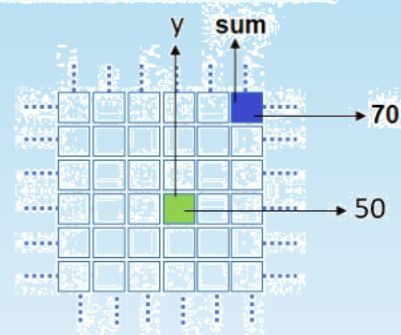


- Biến (variable) đại diện cho vùng nhớ (trên RAM) lưu trữ dữ liệu trong chương trình và có thể thay đổi được

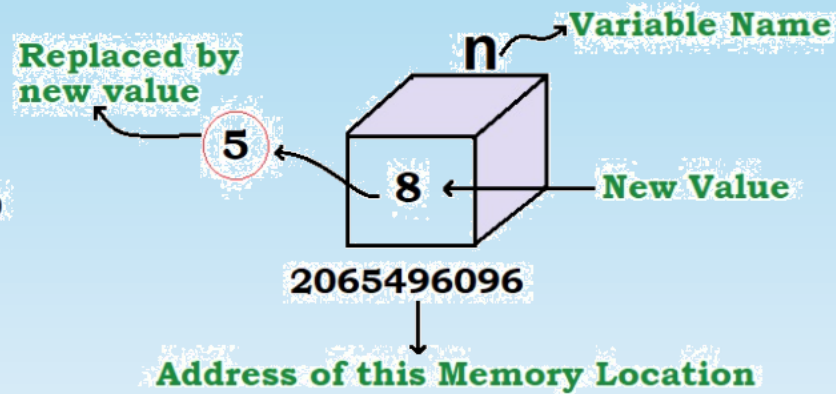
```
x = 20
y = 50
sum = x + y
n = 5
n = 8
```



1<sup>st</sup> Variable created  
>>> x = 20



2<sup>nd</sup> Variable created  
>>> y = 50



- Gán giá trị cho biến sử dụng toán tử gán “=”. Trong python, khi tạo ra một biến phải gán giá trị cho biến đó
- Hằng (constant) là loại biến có giá trị không thể thay đổi được

- Ghi chú lệnh cẩn thận thể hiện tính chuyên nghiệp của lập trình viên
- Viết ghi chú hỗ trợ chính lập trình viên trong việc đọc lại mã nguồn, bảo trì
- Sử dụng ký hiệu hash (#) ghi chú một dòng
- Sử dụng 3 cặp nháy đơn `'''`, 3 cặp nháy đôi `"""` để ghi chú nhiều dòng
- Docstring (documentation string) là chuỗi xuất hiện ngay sau khi định nghĩa một phương thức, lớp hoặc module. Sử dụng dấu nháy `"""` để viết docstring

```
def double(num):  
    """Function to double the value"""  
    return 2*num  
print(double.__doc__)
```

- Một hoặc nhiều câu lệnh (statement) có thể tạo thành một khối lệnh (code block). Python sử dụng thụt đầu dòng (indentation) để bắt đầu định nghĩa và phân tách với các code block khác.

```
"""  
Giải phương trình bậc 1: ax+b=0  
Có 3 trường hợp để biện luận  
Nếu hệ số a =0 và hệ số b=0 ==> vô số nghiệm  
Nếu hệ số a =0 và hệ số b !=0 ==> vô nghiệm  
Nếu hệ số a !=0 ==> có nghiệm -b/a  
"""  
  
a = 0  
b = 113  
if a == 0 and b == 0:  
    print("Vô số nghiệm")  
elif a == 0 and b != 0:  
    print("Vô nghiệm")  
else:  
    print("Có No X=", -b/a)
```



- Toán tử số học - Arithmetic Operators
- Toán tử quan hệ - Comparison (Relational) Operators
- Toán tử gán - Assignment Operators
- Toán tử logic - Logical Operators
- Toán tử Bitwise - Bitwise Operators
- Toán tử khai thác - Membership Operators
- Toán tử xác thực - Identity Operators



| 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    |





| Toán tử            | Mô tả                     | Ví dụ                   |                 |
|--------------------|---------------------------|-------------------------|-----------------|
| <code>==</code>    | So sánh bằng              | <code>5 == 5</code>     | ➔ kết quả True  |
| <code>!=</code>    | So sánh không bằng        | <code>5 != 5</code>     | ➔ kết quả False |
| <code>&lt;</code>  | So sánh nhỏ hơn           | <code>5 &lt; 5</code>   | ➔ kết quả False |
| <code>&lt;=</code> | So sánh nhỏ hơn hoặc bằng | <code>5 &lt;= 5</code>  | ➔ kết quả True  |
| <code>&gt;</code>  | So sánh lớn hơn           | <code>5 &gt; 5.5</code> | ➔ kết quả False |
| <code>&gt;=</code> | So sánh lớn hơn hoặc bằng | <code>8 &gt;= 5</code>  | ➔ kết quả True  |





| Toán tử | Mô tả                                           | Ví dụ                                    | Tương đương  |
|---------|-------------------------------------------------|------------------------------------------|--------------|
| =       | Gán giá trị bên phải cho biến bên trái dấu bằng | $x = 5$                                  |              |
| +=      | Cộng và gán                                     | $x = 2, x += 5 \rightarrow x = 7$        | $x = x + 5$  |
| -=      | Trừ và gán                                      | $x = 2, x -= 5 \rightarrow x = -3$       | $x = x - 5$  |
| *=      | Nhân và gán                                     | $x = 2, x *= 5 \rightarrow x = 10$       | $x = x * 5$  |
| /=      | Chia và gán                                     | $x = 7, x /= 5 \rightarrow x = 1.4$      | $x = x / 5$  |
| //=     | Chia và gán (lấy nguyên)                        | $x = 7, x //= 5 \rightarrow x = 1$       | $x = x // 5$ |
| %=      | Chia lấy dư                                     | $x = 7, x %= 5 \rightarrow x = 2$        | $x = x \% 5$ |
| **=     | Lấy lũy thừa và gán                             | $x = 2, x **= 3 \rightarrow x = 2^3 = 8$ | $x = x ** 3$ |



| Toán tử    | Mô tả                                                                     | Ví dụ                                                                             |
|------------|---------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| <b>and</b> | Cả hai điều kiện là True<br>→ True                                        | x = 2016<br>print(x % 4 == 0 <b>and</b> x % 100 != 0)<br>→ True                   |
| <b>or</b>  | Chỉ cần một điều kiện True<br>Tất cả điều kiện False<br>→ True<br>→ False | x = 2016<br>print((x % 4 == 0 and x % 100 != 0) <b>or</b> x % 400 == 0)<br>→ True |
| <b>not</b> | Đảo ngược trạng thái logic của toán hạng                                  | x = 4<br>if ( <b>not</b> x >= 5):<br>print("Chưa đạt")<br>else:<br>print("Đậu")   |



|                 |                                                           |                            |
|-----------------|-----------------------------------------------------------|----------------------------|
| <b>a</b>        | 12 (00001100)                                             |                            |
| <b>b</b>        | 15 (00001111)                                             |                            |
| <b>Toán tử</b>  | <b>Mô tả</b>                                              | <b>Ví dụ</b>               |
| <b>&amp;</b>    | AND                                                       | $(a \& b) = 12$ (00001100) |
| <b> </b>        | OR                                                        | $(a   b) = 14$ (00001111)  |
| <b>^</b>        | XOR (2 bit giống nhau trả về 0, 2 bit khác nhau trả về 1) | $(a ^ b) = 3$ (00000011)   |
| <b>~</b>        | NOT                                                       | $(-a) = -13$ (00001101)    |
| <b>&lt;&lt;</b> | Phép toán dịch bit trái                                   | $a << 2 = 48$ (00110000)   |
| <b>&gt;&gt;</b> | Phép toán dịch bit phải                                   | $a >> 2 = 3$ (00000011)    |



|                |                                         |                     |                    |
|----------------|-----------------------------------------|---------------------|--------------------|
| <b>a</b>       | 4                                       |                     |                    |
| <b>b</b>       | [1, 5, 7, 6, 9]                         |                     |                    |
| <b>Toán tử</b> | <b>Mô tả</b>                            |                     | <b>Ví dụ</b>       |
| <b>in</b>      | Nếu 1 đối số thuộc một tập đối số       | ➔ True và ngược lại | a in b // False    |
| <b>not in</b>  | Nếu 1 đối số không thuộc một tập đối số | ➔ True và ngược lại | a not in b // True |



|                |                                                      |                       |         |
|----------------|------------------------------------------------------|-----------------------|---------|
| <b>a</b>       | 4                                                    |                       |         |
| <b>b</b>       | 5                                                    |                       |         |
| <b>Toán tử</b> | <b>Mô tả</b>                                         | <b>Ví dụ</b>          |         |
| <b>is</b>      | Toán tử này sẽ trả về True nếu $a == b$ và ngược lại | $a \text{ is } b$     | //False |
| <b>not is</b>  | Toán tử này sẽ trả về True nếu $a != b$ và ngược lại | $a \text{ is not } b$ | //True  |



- Python ràng buộc thứ tự ưu tiên của các toán tử, hoặc dùng ngoặc tròn ( ... ) để thể hiện rõ ràng.

| Thứ tự ưu tiên | Toán tử                  | Miêu tả                                         |
|----------------|--------------------------|-------------------------------------------------|
| 1              | **                       | Toán tử mũ                                      |
| 2              | * / % //                 | Phép nhân, chia, lấy phần dư và lấy phần nguyên |
| 3              | + -                      | Toán tử cộng, trừ                               |
| 4              | <= < > >=                | Các toán tử so sánh                             |
| 5              | <> == !=                 | Các toán tử so sánh                             |
| 6              | = %= /= //= -= += *= **= | Các toán tử gán                                 |
| 7              | is, is not               | Các toán tử so sánh                             |
| 8              | not, or, and             | Các toán tử logic                               |





- Sử dụng hàm `input()` và mang dữ liệu kiểu chuỗi

```
print("Mời bạn nhập:")
s = input()
print("Bạn đã nhập:", s)
print("Kiểu dữ liệu:", type(s))
```

```
x = input("Mời bạn nhập giá trị gì đó:")
print("Bạn nhập: ", x)
```

- Chuyển đổi kiểu dữ liệu

- Số nguyên : `int(input())`

- Số thực : `float(input())`

- Số Boolean: 

```
def StrToBool(s):
    return s.lower() in ("yes", "true", "t", "1")
```

```
print("Mời thím nhập bool:")
x = StrToBool(input())
print("Bạn nhập:", x)
print("Kiểu dữ liệu:", type(x))
```



- Xuất lặp dữ liệu `print('*' * 15)` # xuất dấu '\*' 15 lần
- Dùng hàm format `print('{0}{1}'.format(7, 10 ** 7))`

$10000000$   
`'{0} {1}'.format(7, 10**7)` → `'7 10000000'`  
 $7$

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



- Xuất căn lề phải dữ liệu

```
print('-'*15)
print('{0:>2} {1:>11}'.format('STT', 'Giá trị'))
print('-'*15)
print('{0:>2} {1:>11}'.format(1, 10**10))
print('{0:>2} {1:>11}'.format(2, 10**9))
print('{0:>2} {1:>11}'.format(3, 10**8))
print('{0:>2} {1:>11}'.format(4, 10**7))
print('{0:>2} {1:>11}'.format(5, 10**6))
print('{0:>2} {1:>11}'.format(6, 10**5))
print('{0:>2} {1:>11}'.format(7, 10**4))
print('{0:>2} {1:>11}'.format(8, 10**3))
print('{0:>2} {1:>11}'.format(9, 10**2))
print('{0:>2} {1:>11}'.format(10, 10**1))
print('-'*15)
```



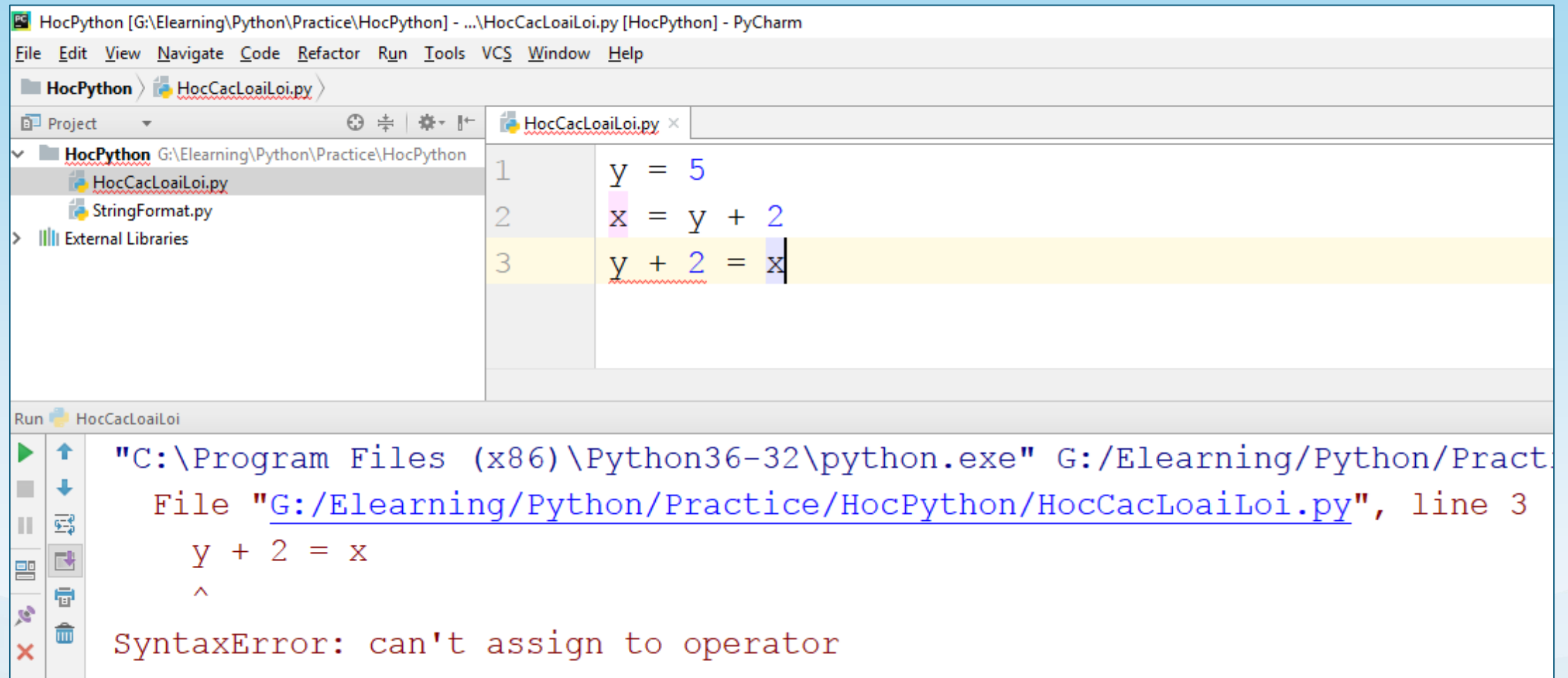
| ----- |             |
|-------|-------------|
| STT   | Giá trị     |
| ----- |             |
| 1     | 10000000000 |
| 2     | 1000000000  |
| 3     | 100000000   |
| 4     | 10000000    |
| 5     | 1000000     |
| 6     | 100000      |
| 7     | 10000       |
| 8     | 1000        |
| 9     | 100         |
| 10    | 10          |
| ----- |             |



- Lỗi cú pháp (Syntax Errors)
- Lỗi thực thi (Run-time Exceptions)
- Lỗi nghiệp vụ (Logic Errors)
- Python cho phép bắt lỗi bằng khối lệnh try ... catch ...



- Lỗi thường gặp với programmer mới bắt đầu học



The screenshot shows the PyCharm IDE interface. The top toolbar includes menus like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project browser on the left shows a project named 'HocPython' with files 'HocCacLoaiLoi.py' and 'StringFormat.py'. The main editor window displays the code in 'HocCacLoaiLoi.py':

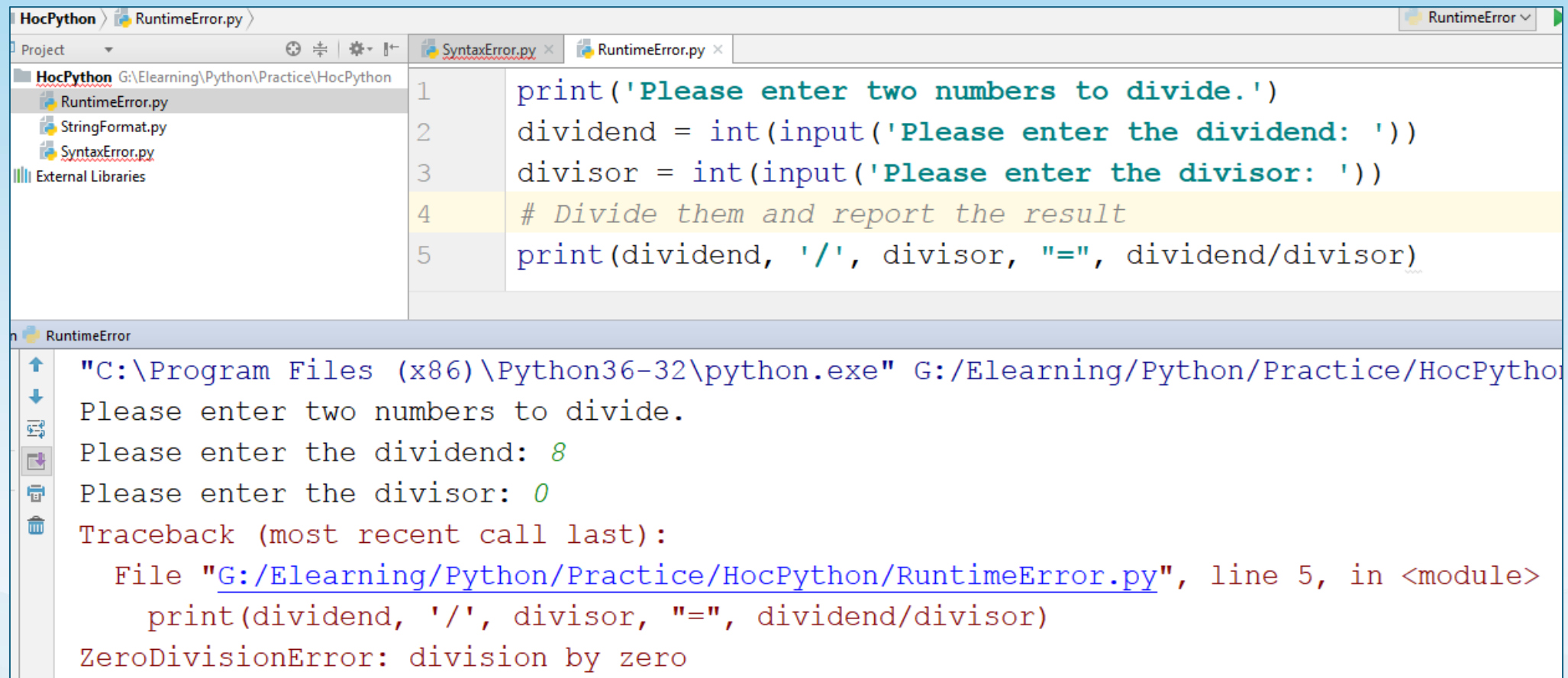
```
1 y = 5
2 x = y + 2
3 y + 2 = x
```

The third line is highlighted in yellow, and a red squiggly line under 'y + 2' indicates an error. The bottom panel shows the Run output with the following message:

```
"C:\Program Files (x86)\Python36-32\python.exe" G:/Elearning/Python/Pract.
File "G:/Elearning/Python/Practice/HocPython/HocCacLoaiLoi.py", line 3
    y + 2 = x
    ^
SyntaxError: can't assign to operator
```



- Lỗi này thường thông dịch hoàn tất nhưng quá trình thực thi lại phát sinh lỗi.
- Một số lỗi thường gặp như: chia cho 0, ép sai kiểu, đọc file lỗi, kết nối mạng lỗi,...



The screenshot shows a Python IDE window titled 'HocPython' with a file named 'RuntimeError.py'. The code in the editor is as follows:

```
1 print('Please enter two numbers to divide.')
2 dividend = int(input('Please enter the dividend: '))
3 divisor = int(input('Please enter the divisor: '))
4 # Divide them and report the result
5 print(dividend, '/', divisor, "=", dividend/divisor)
```

The bottom pane shows the error message:

```
RuntimeError
"C:\Program Files (x86)\Python36-32\python.exe" G:/Elearning/Python/Practice/HocPython
Please enter two numbers to divide.
Please enter the dividend: 8
Please enter the divisor: 0
Traceback (most recent call last):
  File "G:/Elearning/Python/Practice/HocPython/RuntimeError.py", line 5, in <module>
    print(dividend, '/', divisor, "=", dividend/divisor)
ZeroDivisionError: division by zero
```





- Lỗi này rất nghiêm trọng và không có biểu hiện cụ thể rõ ràng, hậu quả kết quả sai với yêu cầu của khách hàng.
- Lỗi này thường rất hiếm xảy ra do programmer chưa hiểu rõ đặc tả chức năng, tích hợp hệ thống  
→ rất khó phát hiện và khó fix bug



- Python hỗ trợ try ... catch để bắt lỗi Runtime

```
HandleException.py x
1  try:
2      x=5
3      y=0
4      z=x/y
5  except: # Catch absolutely any other type of exception
6      print('This program has encountered a problem')
```



- ✓ Họ tên : **Trần Quang Khải**
- ✓ Email : **khaitq@hcmute.edu.vn**
- ✓ Zalo (mã Qr)

