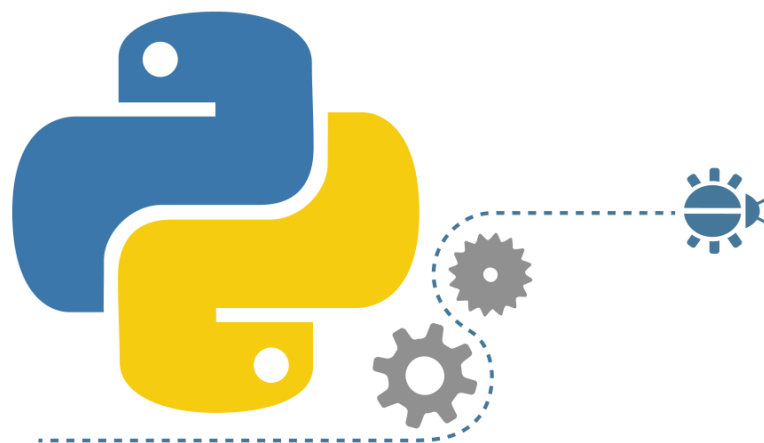


Cách nhập liệu từ bàn phím trong Python



Nội dung bài học

Trong Python để nhập liệu từ bàn phím ta dùng hàm `input()`. Giá trị nhập vào của hàm `input()` thường là kiểu chuỗi, do đó ta cần chuyển kiểu nếu như muốn lưu trữ giá trị nhập vào không phải kiểu chuỗi.

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

Nội dung bài học

Muốn đưa về số int

```
print ("Mời bạn nhập 1 số nguyên: ")  
x=int(input() )  
print ("Bạn nhập: ",x)  
print ("Kiểu dữ liệu: ",type(x) )
```

Nội dung bài học

Muốn đưa về số float

```
print("Mời bạn nhập 1 số thực:")  
x=float(input())  
print("Bạn nhập:", x)  
print("Kiểu dữ liệu:", type(x))
```

Nội dung bài học

Muốn đưa về số boolean

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

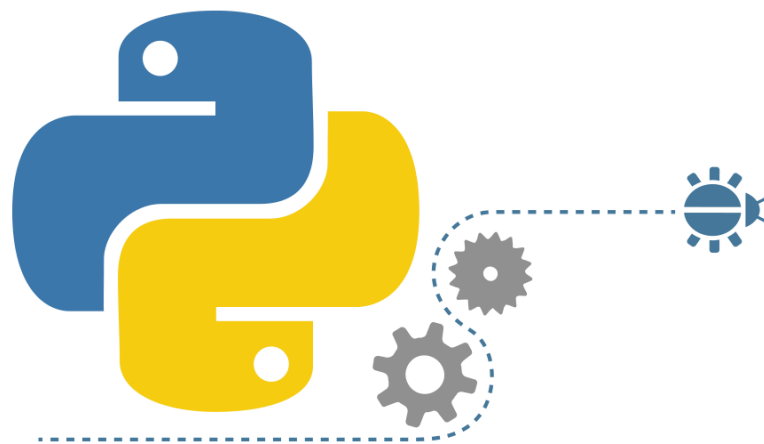
```
print("Mời bạn nhập 1 số kiểu bool:")  
x = StrToBool(input())  
print("Bạn nhập:", x)  
print("Kiểu dữ liệu:", type(x))
```

Nội dung bài học

Ngoài ra hàm `input()` còn có cho phép ta nhập nhãn tiêu đề vào như sau:

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

Các kiểu xuất dữ liệu



Nội dung bài học

Để xuất lặp dữ liệu ra màn hình ta làm như sau:

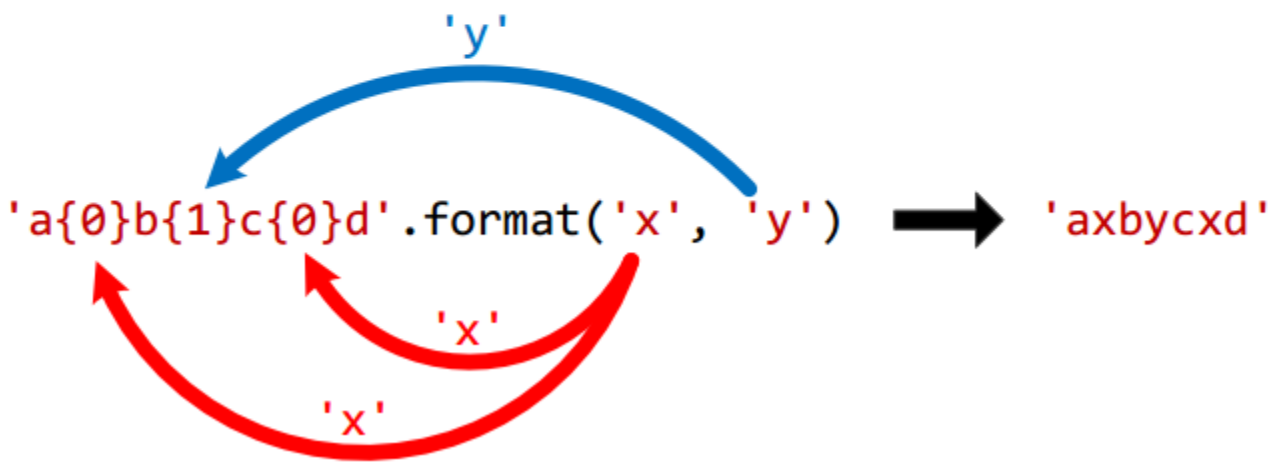
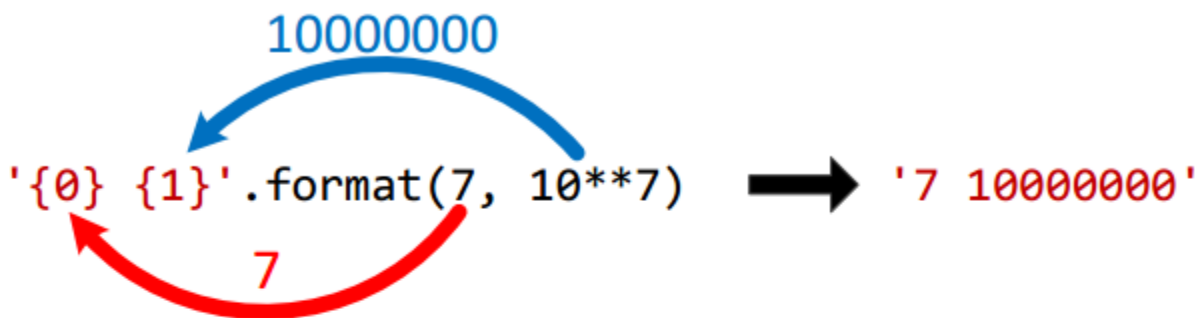
```
print ( '*' * 15 )  
*****
```

➔ Xuất dấu * 15 lần

Nội dung bài học

Dùng hàm format để xuất dữ liệu:

```
print('{0} {1}'.format(7, 10**7))
```



Nội dung bài học

Muốn xuất căn phải ta có thể làm như sau:

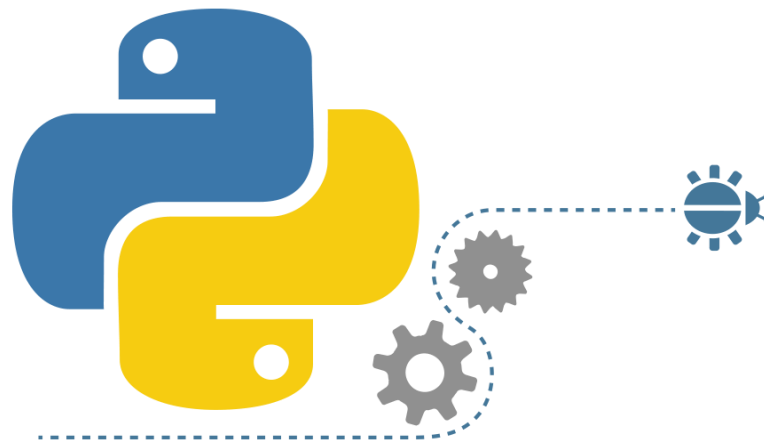
```
print('-'*15)
print('{0:>2} {1:>11}'.format('STT', 'Giá trị'))
print('-'*15)
print('{0:>2} {1:>11}'.format(1, 10**10))
print('{0:>2} {1:>11}'.format(2, 10**9))
print('{0:>2} {1:>11}'.format(3, 10**8))
print('{0:>2} {1:>11}'.format(4, 10**7))
print('{0:>2} {1:>11}'.format(5, 10**6))
print('{0:>2}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

Các loại lỗi trong Python



Nội dung bài học

Bất cứ người nào lập trình cũng đều sinh ra lỗi, tùy vào và khả năng và kinh nghiệm của từng người mà gặp các loại lỗi khác nhau.

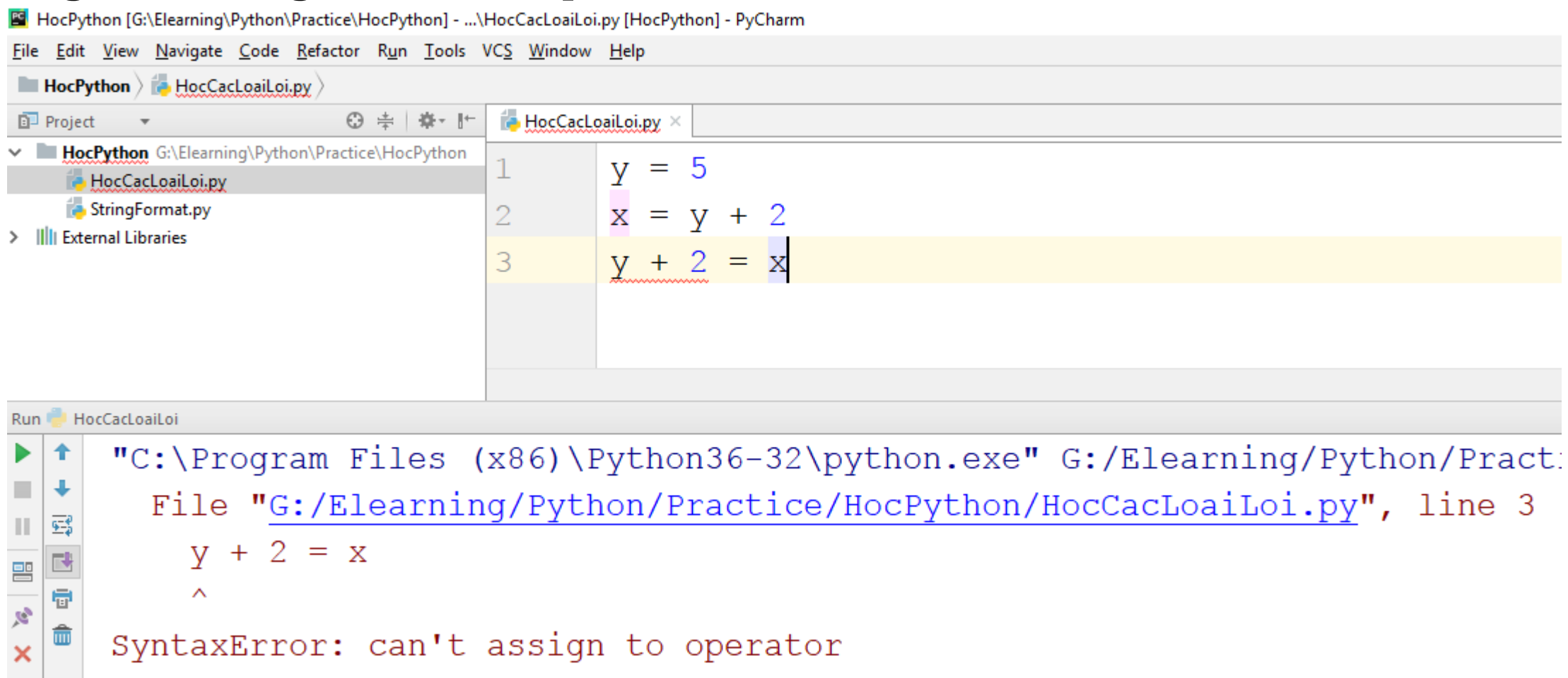
Có 3 loại lỗi thường gặp là:

1. Lỗi cú pháp (Syntax Errors)
2. Lỗi thực thi (Run-time Exceptions)
3. Lỗi nghiệp vụ (Logic Errors)

Python cũng giống như các ngôn ngữ khác là cho phép bắt lỗi bằng khối lệnh try...catch

1. Lỗi cú pháp(Syntax Errors)

Lỗi thường gặp với những Programmer mới bắt đầu học, viết đâu sai đó, không thể thông dịch được phần mềm.



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The project name is 'HocPython' and the file being edited is 'HocCacLoaiLoi.py'. The code editor shows three lines of Python code:

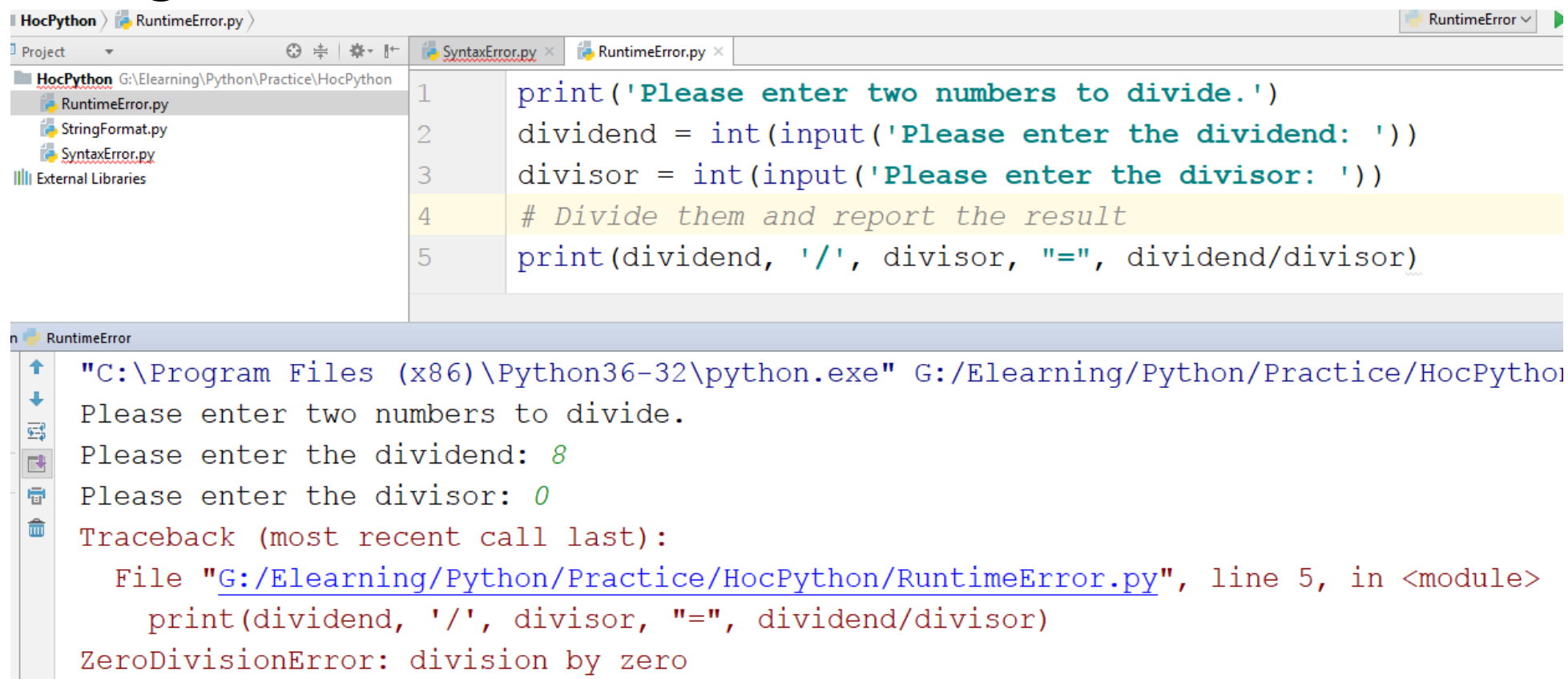
```
1 y = 5
2 x = y + 2
3 y + 2 = x
```

Line 3 is highlighted in yellow, indicating a syntax error. The error message in the Run console at the bottom reads:

```
"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
```

2. 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....



```
HocPython > RuntimeError.py
Project
HocPython G:\Elearning\Python\Practice\HocPython
  RuntimeError.py
  StringFormat.py
  SyntaxError.py
  External Libraries

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)

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
```

3. Lỗi nghiệp vụ (Logic Errors)

Lỗi này rất nghiêm trọng, thông dịch và chạy ngon lành. Nhưng kết quả sai với yêu cầu nghiệp vụ của Khách hàng.

Các lỗi này thường rất hiếm khi xảy ra, nhưng khi xảy ra rồi thì rất khó tìm thấy, và thấy rồi cũng thường khó fix bug.

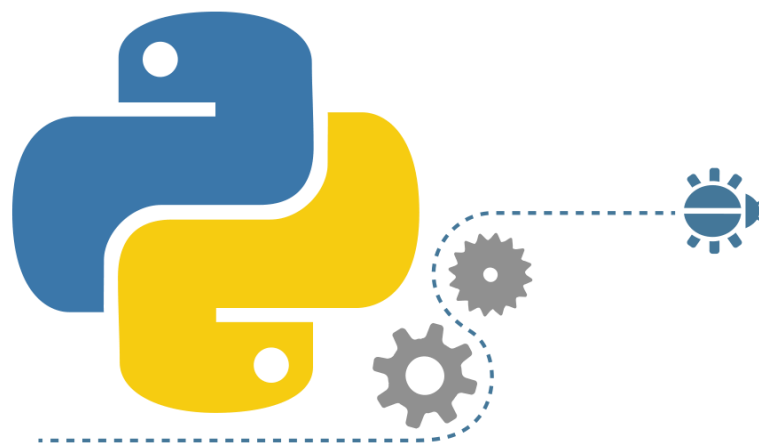
4. Bắt lỗi

Python hỗ trợ **try...catch** để bắt lỗi **Runtime**, giúp báo rõ loại lỗi chương trình đang gặp và vẫn tiếp tục hoạt động khi gặp lỗi. Có rất nhiều trường hợp sử dụng, bài học trình bày 1 cách tổng quát nhất:

```
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')
```


Bài tập rèn luyện

Tính chu vi diện tích Hình tròn



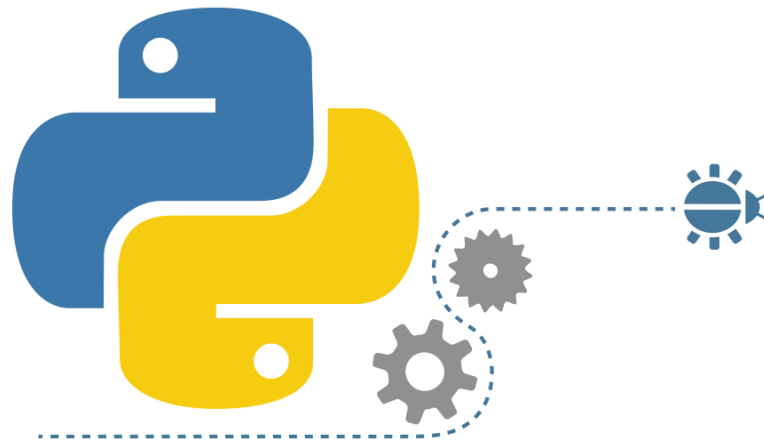
Nội dung bài học

Nhập bán kính đường tròn r .

Tính và xuất chu vi, diện tích đường tròn tương ứng.

HD: $cv = 2 * \pi * r$ và $dt = \pi * r * r$

Bài tập rèn luyện Tính giờ phút giây



Nội dung bài học

Nhập vào số **giây** bất kỳ t . Tính và xuất ra dạng

Giờ:Phút:Giây

Ví dụ: Nhập 3750 thì xuất ra 1:2:30 AM

Nhập 51100 thì xuất ra 2:11:40 PM

HD:

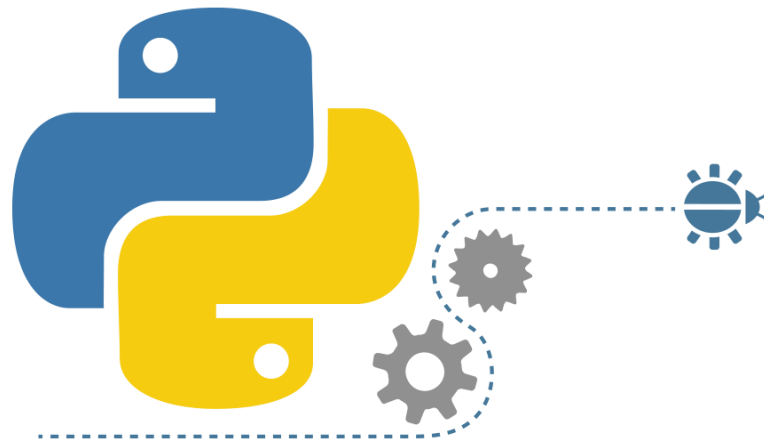
$\text{hour} = (t / 3600) \% 24$

$\text{minute} = (t \% 3600) / 60$

$\text{second} = (t \% 3600) \% 60$

Bài tập rèn luyện

Tính điểm trung bình

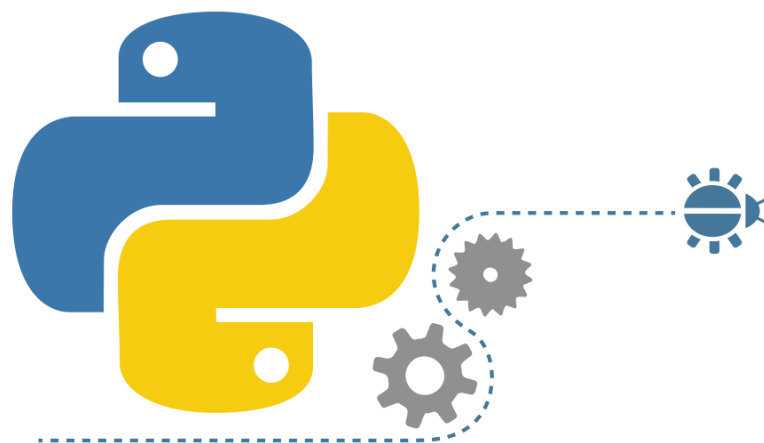


Nội dung bài học

Viết chương trình nhập vào điểm ba môn Toán, Lý, Hóa của một học sinh.

In ra điểm trung bình của học sinh đó với hai số lẻ thập phân.

Các bài tập tự rèn luyện



Nội dung bài học

Câu 1: Python hỗ trợ những kiểu dữ liệu cơ bản nào?

Câu 2: Trình bày các loại ghi chú trong Python

Câu 3: Trình bày ý nghĩa toán tử `/`, `//`, `%`, `**`, `and`, `or`, `is`

Câu 4: Trình bày một số cách nhập dữ liệu từ bàn phím

Câu 5: Trình bày các loại lỗi khi lập trình và cách bắt lỗi trong Python

Nội dung bài học

Câu 6:

Cho các biến với giá trị

$i1 = 2$

$i2 = 5$

$i3 = -3$

$d1 = 2.0$

$d2 = 5.0$

$d3 = -0.5$

Cho biết kết quả của các lệnh sau:

(a) $i1 + (i2 * i3)$

(b) $i1 * (i2 + i3)$

(c) $i1 / (i2 + i3)$

(d) $i1 // (i2 + i3)$

(e) $i1 / i2 + i3$

(f) $i1 // i2 + i3$

(g) $3 + 4 + 5 / 3$

(h) $3 + 4 + 5 // 3$

(i) $(3 + 4 + 5) / 3$

(j) $(3 + 4 + 5) // 3$

(k) $d1 + (d2 * d3)$

(l) $d1 + d2 * d3$

(m) $d1 / d2 - d3$

(n) $d1 / (d2 - d3)$

(o) $d1 + d2 + d3 / 3$

(p) $(d1 + d2 + d3) / 3$

(q) $d1 + d2 + (d3 / 3)$

(r) $3 * (d1 + d2) * (d1 - d3)$

Nội dung bài học

Câu 7: Hãy viết ngắn gọn lại các lệnh dưới đây:

(a) $x = x + 1 \rightarrow x += 1$

(b) $x = x / 2$

(c) $x = x - 1$

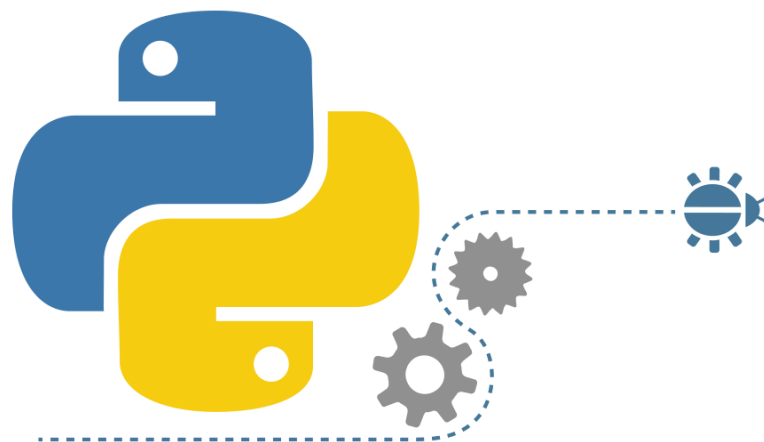
(d) $x = x + y$

(e) $x = x - (y + 7)$

(f) $x = 2 * x$

(g) $\text{number_of_closed_cases} = \text{number_of_closed_cases} + 2 * \text{ncc}$

Biểu thức Boolean



Nội dung bài học

- Biểu thức **Boolean** (**Boolean Expression**) còn được gọi là **Predicate**.
- Là một biểu thức rất quan trọng và phổ biến trong các lệnh của Python cũng như ngôn ngữ lập trình khác.
- Các giá trị là **True** hoặc **False**, dựa vào các giá trị này mà ta điều hướng các công việc trong phần mềm.

Nội dung bài học

Ví dụ:

```
a = True
```

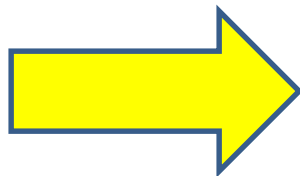
```
b = False
```

```
print('a =', a, ' b =', b)
```

```
# gán lại kết quả cho a
```

```
a = False
```

```
print('a =', a, ' b =', b)
```



```
a = True b = False
```

```
a = False b = False
```

Nội dung bài học

Bảng tổng quát:

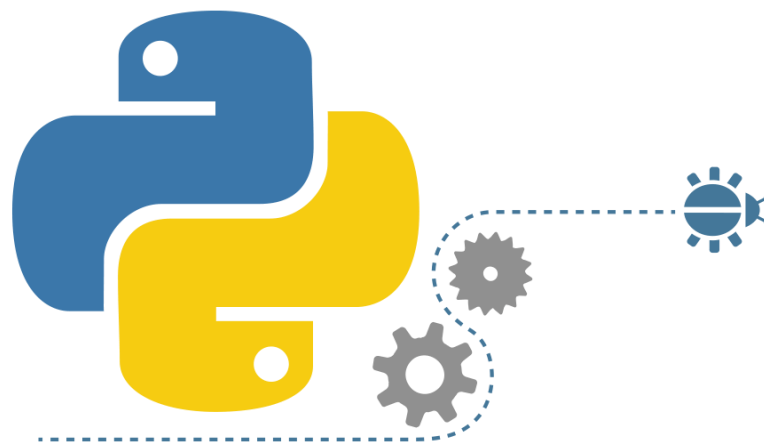
Biểu thức	Ý nghĩa
$x == y$	True nếu $x=y$, False nếu x khác y
$x < y$	True nếu $x < y$, False nếu $x \geq y$
$x \leq y$	True nếu $x \leq y$, False nếu $x > y$
$x > y$	True nếu $x > y$, False nếu $x \leq y$
$x \geq y$	True nếu $x \geq y$, False nếu $x < y$
$x \neq y$	True nếu x khác y , False nếu $x=y$

Nội dung bài học

Ví dụ:

Biểu thức	Ý nghĩa
$10 < 20$	True
$10 \geq 20$	False
$x < 100$	True if x nhỏ hơn 100; ngược lại False

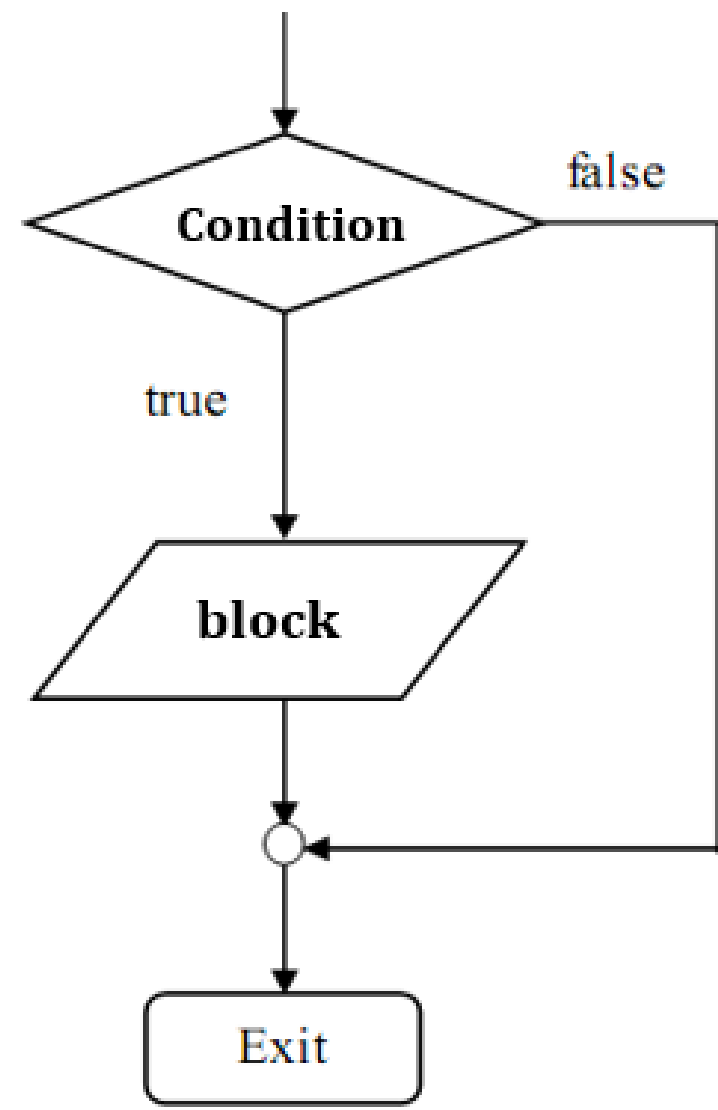
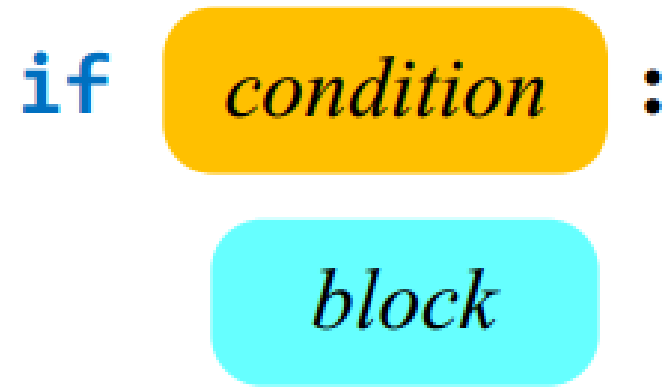
Biểu thức If



Nội dung bài học

Biểu thức **if** là một biểu thức điều kiện rất quan trọng và phổ biến trong Python. Biểu thức **if** đứng một mình chỉ quan tâm tới điều kiện đúng (**True**). Khi điều kiện đúng thì khối lệnh bên trong **if** sẽ được thực thi.

Cú pháp:



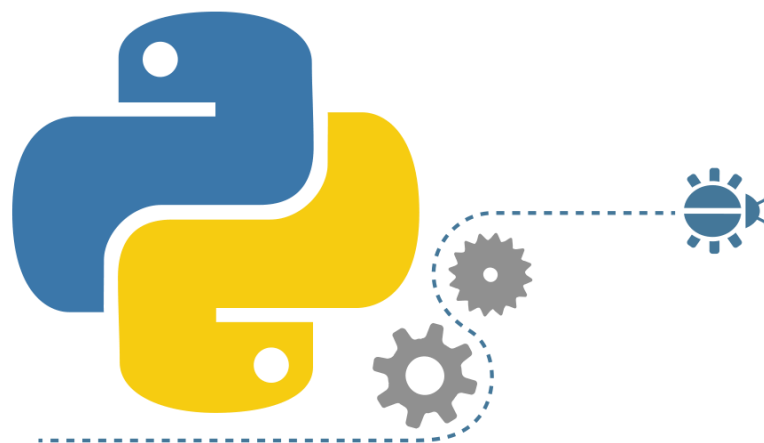
Nội dung bài học

Ví dụ:

```
dtb=float(input("Nhập điểm trung bình:"))  
if dtb>=5:  
    print("Bạn đã đậu!")  
    print("Hú hồn!")
```

Lưu ý rằng Python không dùng ngoặc nhọn để bao bọc các dòng lệnh, mà ta dùng phím Tab hoặc khoảng trắng thụt đầu dòng.

Biểu thức if ... else



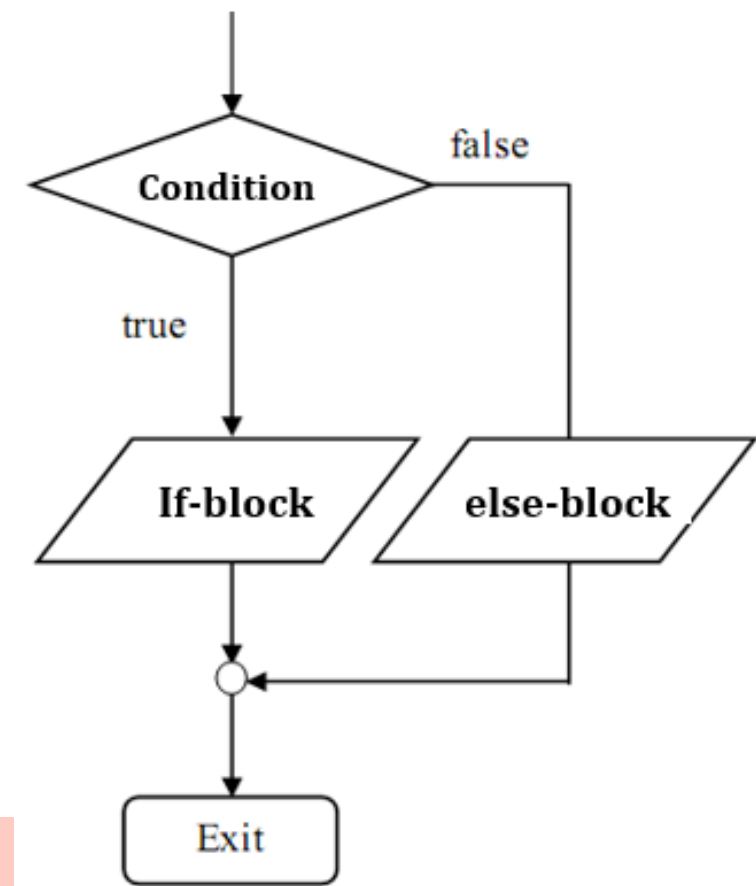
Nội dung bài học

Biểu thức **if...else** là một biểu thức điều kiện rất quan trọng và phổ biến trong **Python**. Biểu thức này quan tâm điều kiện đúng (**True**) và sai (**False**). Nó phổ biến hơn biểu thức **if**.

Cú pháp:

```

if condition :
    if-block
else:
    else-block
    
```

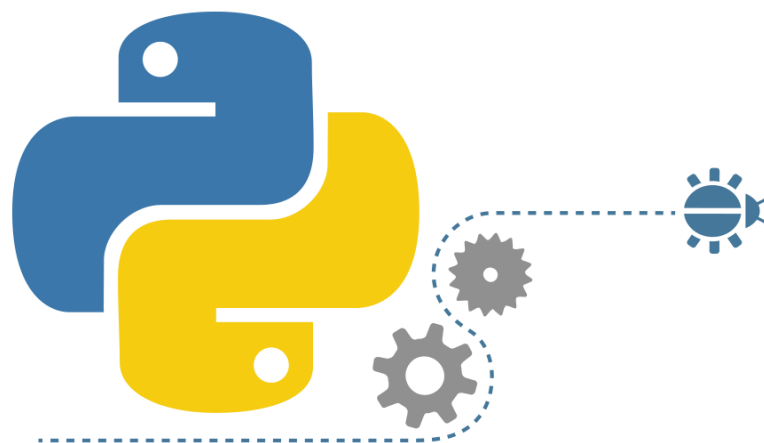


Nội dung bài học

Ví dụ:

```
dtb = float(input("Nhập điểm trung bình:"))  
if dtb >= 5:  
    print("Bạn đã đậu!")  
    print("Hú hồn!")  
else:  
    print("Ở nhà lấy Vợ")  
    print("Đi phụ hồ")
```

Biểu thức If ... elif lồng nhau

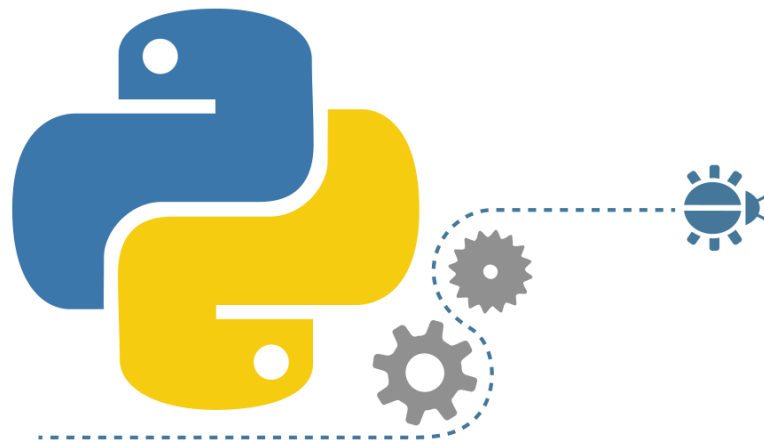


Nội dung bài học

Với các điều kiện thức tạp, Python cũng hỗ trợ kiểm tra điều kiện if elif lồng nhau:

```
dtb = float(input("Nhập điểm trung bình:"))  
if dtb >= 9:  
    print("Bạn xếp loại giỏi")  
elif dtb >= 7:  
    print("Bạn xếp loại khá")  
elif dtb >= 5:  
    print("Bạn xếp loại Trung bình")  
else:  
    print("Chia tay hoàng hôn")
```

Biểu thức pass



Nội dung bài học

Biểu thức **pass** khá lợi hại, nó dùng để dành chỗ lập trình. Ví dụ bạn biết chỗ đó phải viết rất nhiều coding, nhưng tại thời điểm này chưa kịp làm. Ta sẽ dùng **pass** để đánh dấu vị trí đó.

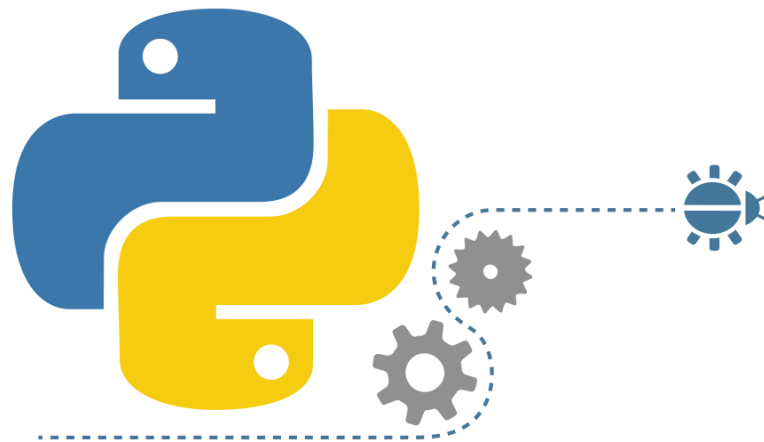
```
a=float(input("Nhập hệ số a:"))
b=float(input("Nhập hệ số b:"))
if a==0:
    else:
        x=-b/a;
        print("{0}x+{1}=0".format(a,b))
        print("có nghiệm x=",x)
```

Lỗi, Python không cho để trống như thế này



```
a=float(input("Nhập hệ số a:"))
b=float(input("Nhập hệ số b:"))
if a==0:
    pass
else:
    x=-b/a;
    print("{0}x+{1}=0".format(a,b))
    print("có nghiệm x=",x)
```

So sánh số thực trong Python



Nội dung bài học

Khi ta thao tác với số thực thì có một chút rắc rối ở chỗ Sai Số, nên ta cần có một ngưỡng Sai Số cho phép (tùy thuộc vào quyết định của người dùng)

```
1 d1 = 1.11 - 1.10
2 d2 = 2.11 - 2.10
3 print('d1 =', d1, ' d2 =', d2)
4 if d1 == d2:
5     print('d1 và d2 bằng nhau')
6 else:
7     print('d1 và d2 khác nhau')
```

Nếu mắt thường quan sát ta tưởng rằng d1 bằng d2 vì lý do sau:

$1.11 - 1.10 \rightarrow 0.01$

$2.11 - 2.10 \rightarrow 0.01$

Nhưng thực ra số thực nó có sai số, nó không phải 100% là 0.01

$d1 = 0.0100000000000000009$ $d2 = 0.009999999999999999787$

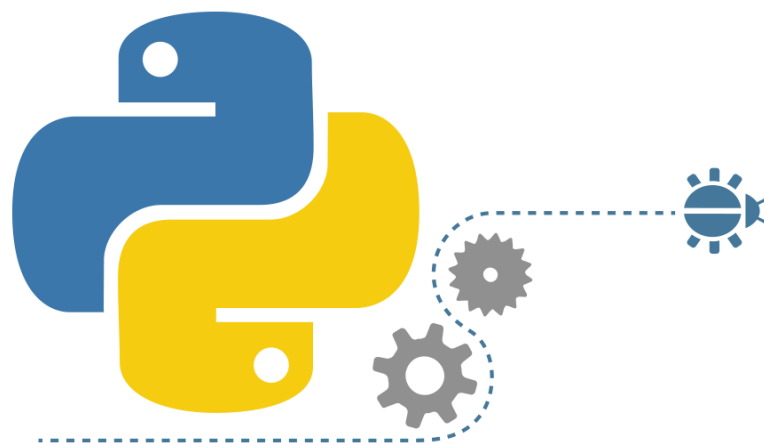
d1 và d2 khác nhau

Nội dung bài học

Do đó ta cho 1 cái ngưỡng so sánh theo sai số cho phép, ví dụ:

```
1 d1 = 1.11 - 1.10
2 d2 = 2.11 - 2.10
3 print('d1 =', d1, ' d2 =', d2)
4 diff = d1 - d2 # Compute difference
5 if diff < 0: # Compute absolute value
6     diff = -diff
7 if diff < 0.0000001: # Are the values close enough?
8     print('Same')
9 else:
10    print('Different')
```

Sử dụng if else như phép gán



Nội dung bài học

Đôi khi việc thực hiện if else trong một biểu thức quá đơn giản sẽ làm cho if else phức tạp không cần thiết:

```

a=5
b=7
if a != b:
    c = 113
else:
    c = 115
print(c)

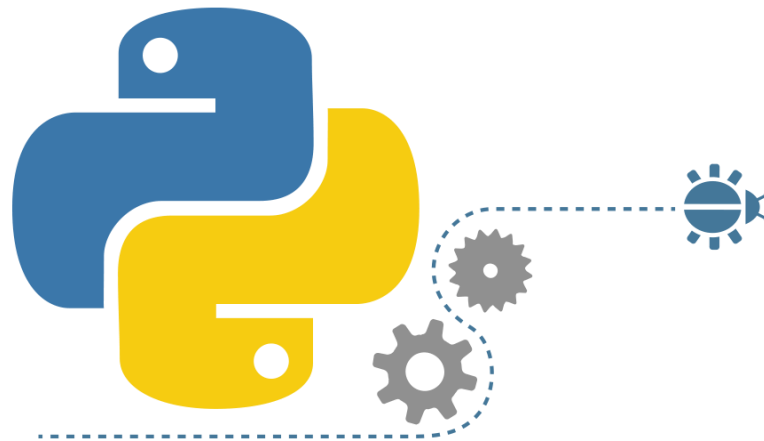
```

→

c = 113
if a != b
else 115



Bài tập rèn luyện Kiểm tra năm nhuận



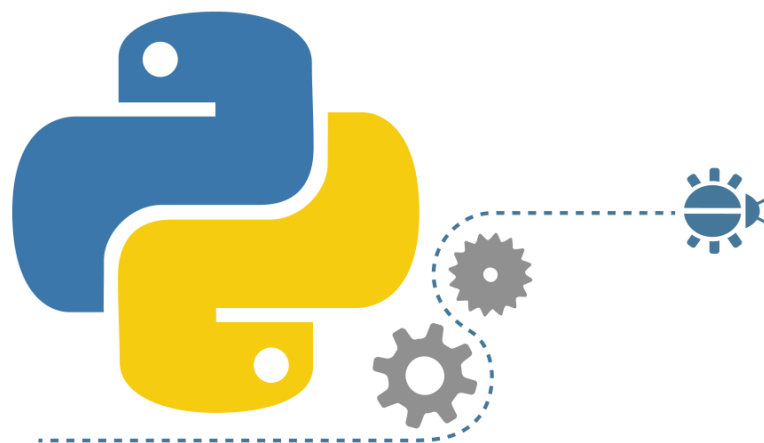
Nội dung bài học

Nhập vào một năm bất kỳ, kiểm tra năm đó có phải năm nhuận hay không.

Biết rằng: Năm nhuận là năm chia hết cho 4 nhưng không chia hết cho 100 hoặc chia hết cho 400

Bài tập rèn luyện

Đếm số ngày trong tháng



Nội dung bài học

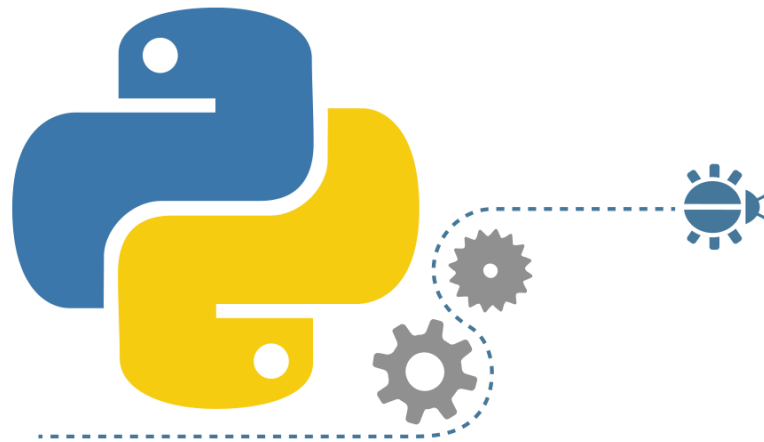
Nhập vào 1 tháng, xuất tháng đó có bao nhiêu ngày.

1,3,5,7,8,10,12 → 31 ngày

4,6,9,11 → có 30 ngày

Nếu là tháng 2 thì yêu cầu nhập thêm năm. Năm nhuận thì tháng 2 có 29 ngày, không nhuận có 28 ngày

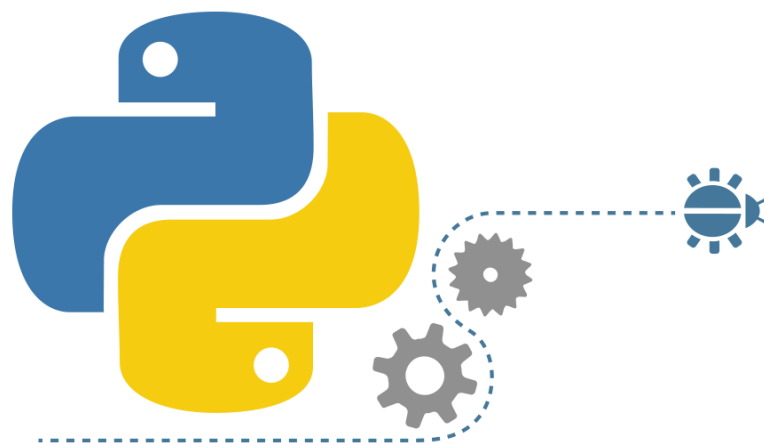
Bài tập rèn luyện Phương trình bậc 2



Nội dung bài học

Viết chương trình giải phương trình bậc 2:
 $ax^2+bx+c=0$

Các bài tập tự rèn luyện



Nội dung bài học

Câu 1:

Cho $x, y, z = 3, 5, 7$. Hãy cho biết kết quả của Boolean Expression

- | | |
|-----------------------------|----------------------------|
| (a) $x == 3$ | (h) $x < 0$ and $x < 10$ |
| (b) $x < y$ | (i) $x \geq 0$ and $x < 2$ |
| (c) $x \geq y$ | (j) $x < 0$ or $x < 10$ |
| (d) $x \leq y$ | (k) $x > 0$ or $x < 10$ |
| (e) $x \neq y - 2$ | (l) $x < 0$ or $x > 10$ |
| (f) $x < 10$ | |
| (g) $x \geq 0$ and $x < 10$ | |

Nội dung bài học

Câu 2:

Cho i, j, k là các con số và lệnh dưới đây:

```
if i < j:
    if j < k:
        i = j
    else:
        j = k
else:
    if j > k:
        j = i
    else:
        i = k
print("i =", i, " j =", j, " k =", k)
```

Hãy cho biết kết quả xuất ra màn hình nếu tuần tự 3 biến trên có các giá trị sau:

- (a) $i = 3, j = 5$, and $k = 7$
- (b) $i = 3, j = 7$, and $k = 5$
- (c) $i = 5, j = 3$, and $k = 7$
- (d) $i = 5, j = 7$, and $k = 3$
- (e) $i = 7, j = 3$, and $k = 5$
- (f) $i = 7, j = 5$, and $k = 3$

Nội dung bài học

Câu 3: Nhập một số n có tối đa 2 chữ số. Hãy cho biết cách đọc ra dạng chữ.

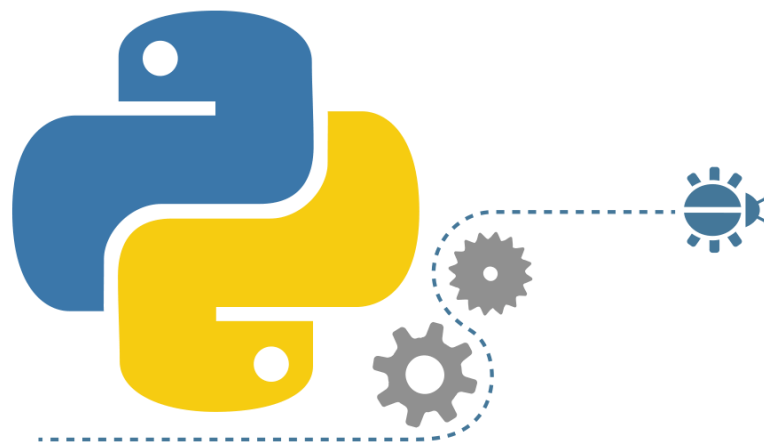
(vd: $n=35 \Rightarrow$ Ba mươi lăm, $n=5 \Rightarrow$ năm).

Câu 4: Nhập vào một ngày (ngày, tháng, năm). Tìm ngày kế sau ngày vừa nhập (ngày/tháng/năm).

Câu 5: Nhập vào 2 giá trị a, b và phép toán '+', '-', '*', '/'. Hãy xuất kết quả theo đúng phép toán đã nhập.

Câu 6: Nhập vào 1 tháng, xuất ra tháng đó thuộc quý mấy trong năm.

Vòng lặp while



Nội dung bài học

While dùng để yêu cầu 1 công việc được lặp đi lặp lại

Nếu **condition** là **True** thì **block** sẽ được lặp lại

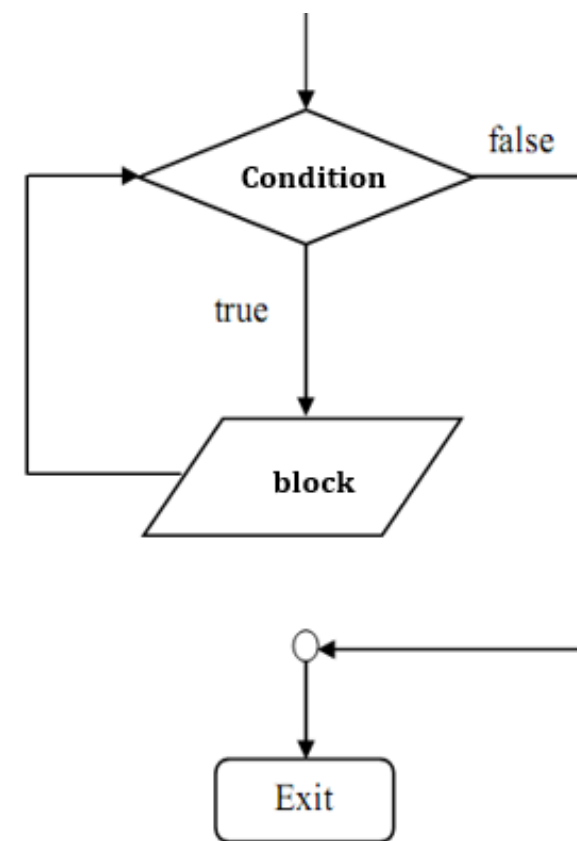
Cú pháp:

while *condition* :

block

-Có thể **block** sẽ không được thực hiện lần nào nếu **condition** là **False** ngay từ đầu

-Ta có thể kết thúc vòng **while** bằng cách đưa **condition** về **False** hoặc dùng từ khóa **break** để thoát.



Nội dung bài học

Ví dụ:

Viết chương trình yêu cầu nhập vào một số nguyên dương [1..10], nếu nhập sai yêu cầu nhập lại. Khi nhập đúng thì xuất ra bình phương của giá trị mới nhập vào.

```
1 value=-1;  
2 while value < 1 or value > 10:  
3     value=int(input("Nhập giá trị [1..10]:"))  
4 print("value=",pow(value,2));
```

```

1  #s=1+2+3+...+N
2  print("Nhập N:")
3  n=int(input())
4  s=0
5  i=1
6  while i<=n:
7      s=s+i
8      i=i+1
9  print("Tổng =",s)

```

Khởi tạo: $s=0, i=1, n=5$

1) $i \leq n \Leftrightarrow 1 \leq 5 \rightarrow \text{True}$

$s=s+i=0+1=1; i=i+1=1+1=2$

2) $i \leq n \Leftrightarrow 2 \leq 5 \rightarrow \text{True}$

$s=s+i=1+2=3; i=i+1=2+1=3$

3) $i \leq n \Leftrightarrow 3 \leq 5 \rightarrow \text{True}$

$s=s+i=3+3=6; i=i+1=3+1=4$

4) $i \leq n \Leftrightarrow 4 \leq 5 \rightarrow \text{True}$

$s=s+i=6+4=10; i=i+1=4+1=5$

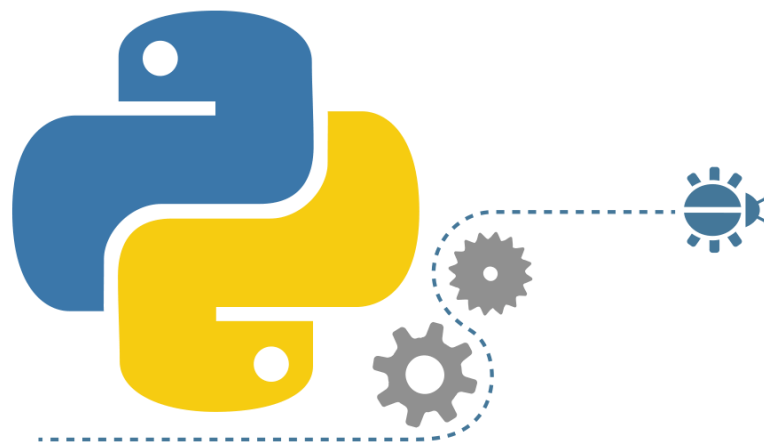
5) $i \leq n \Leftrightarrow 5 \leq 5 \rightarrow \text{True}$

$s=s+i=10+5=15; i=i+1=5+1=6$

6) $i \leq n \Leftrightarrow 6 \leq 5 \rightarrow \text{False} \rightarrow$ Dừng while

\rightarrow xuất tổng = 15

Vòng lặp for



Nội dung bài học

for dùng để lặp tuần tự các công việc, **for** sử dụng **range** để định nghĩa vùng dữ liệu lặp và bước lặp

Cú pháp hàm range:

begin: Giá trị bắt đầu

end: Giá trị cuối

step: Bước nhảy

range(begin, end, step)

Ví dụ cách hoạt động của range:

- `range(10)` → 0; 1; 2; 3; 4; 5; 6; 7; 8; 9
- `range(1, 10)` → 1; 2; 3; 4; 5; 6; 7; 8; 9
- `range(1, 10, 2)` → 1; 3; 5; 7; 9
- `range(10, 0, -1)` → 10; 9; 8; 7; 6; 5; 4; 3; 2; 1
- `range(10, 0, -2)` → 10; 8; 6; 4; 2
- `range(2, 11, 2)` → 2; 4; 6; 8; 10

Nội dung bài học

Các Ví dụ về for:

```
for n in range(10): —————> 0 1 2 3 4 5 6 7 8 9
    print(n, end=' ')
```

```
for n in range(1, 10): —————> 1 2 3 4 5 6 7 8 9
    print(n, end=' ')
```

```
for n in range(1, 10, 2): —————> 1 3 5 7 9
    print(n, end=' ')
```

```
for n in range(10, 0, -1): —————> 10 9 8 7 6 5 4 3 2 1
    print(n, end=' ')
```

```
for n in range(10, 0, -2): —————> 10 8 6 4 2
    print(n, end=' ')
```

```
for n in range(2, 11, 2): —————> 2 4 6 8 10
    print(n, end=' ')
```

```
1 n=int(input("Mời nhập số:"))
2 s=0
3 if n % 2==0:
4     for x in range(2,n+1,2):
5         s=s+x
6 elif n%2!=0:
7     for x in range(1,n+1,2):
8         s=s+x
9 print("Tổng s=",s)
```

N=8, s=0

N=8 là số chẵn $n\%2==0 \Rightarrow 8\%2=0$

1) $x=2 \Rightarrow s=s+x=0+2=2$

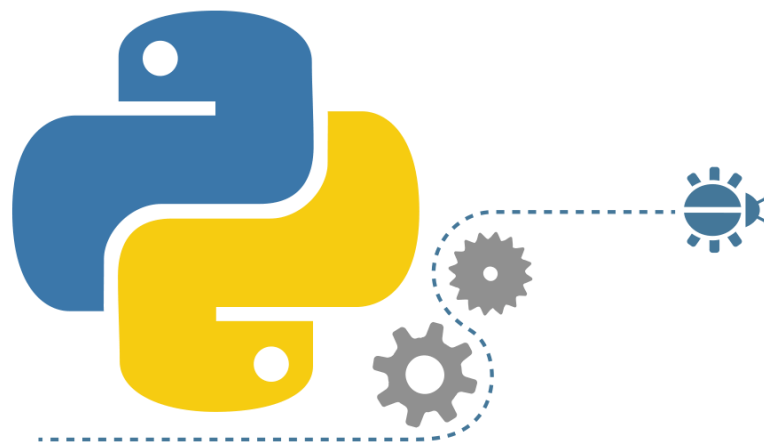
2) $x=4 \Rightarrow s=s+x=2+4=6$

3) $x=6 \Rightarrow s=s+x=6+6=12$

4) $x=8 \Rightarrow s=s+x=12+8=20$

5) $x=10 \Rightarrow$ thấy $10 > n+1=9$

Câu lệnh break



Nội dung bài học

break là từ khóa đặc biệt dùng để thoát khỏi vòng lặp chứa nó trực tiếp khi đạt được mức yêu cầu nào đó.

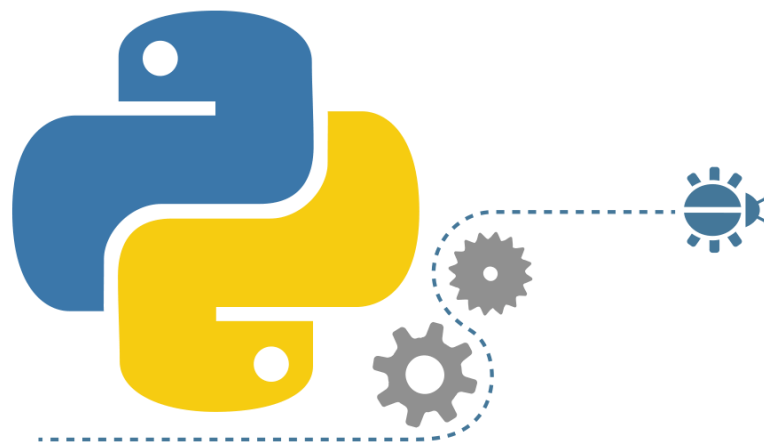
Gặp lệnh **break**, chương trình sẽ không thực hiện bất cứ lệnh nào bên dưới nó, mà thoát luôn khỏi vòng lặp.

Nội dung bài học

Ví dụ: Viết chương trình vòng lặp vĩnh cửu cho phép phần mềm chạy liên tục, khi nào hỏi thoát mới thoát phần mềm:

```
while True:
    a=int(input("Nhập giá trị:"))
    print("Giá trị bạn nhập ",a)
    s=input("Tiếp tục phần mềm không? (c/k) :")
    if s=="c":
        break
print("BYE!")
```

Câu lệnh continue



Nội dung bài học

continue là từ khóa đặc biệt dùng để nhảy sớm tới lần lặp kế tiếp, các lệnh bên dưới **continue** sẽ không được thực thi.

Lưu ý khác với **break**, gặp **break** thì ngừng luôn vòng lặp, gặp **continue** chỉ dừng lần lặp hiện tại đang dở dang để chuyển qua lần lặp tiếp theo.

Nội dung bài học

Ví dụ:

Tính tổng các chữ số lẻ từ 1->15, ngoại trừ số 3 và số 11

```
sum=0
for n in range(1,16,2):
    if n is 3 or n is 11:
        continue
    sum+=n
print(sum)
```

END

