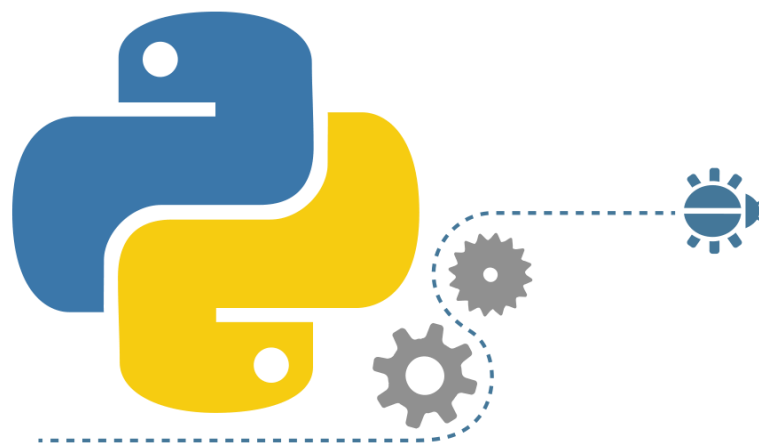


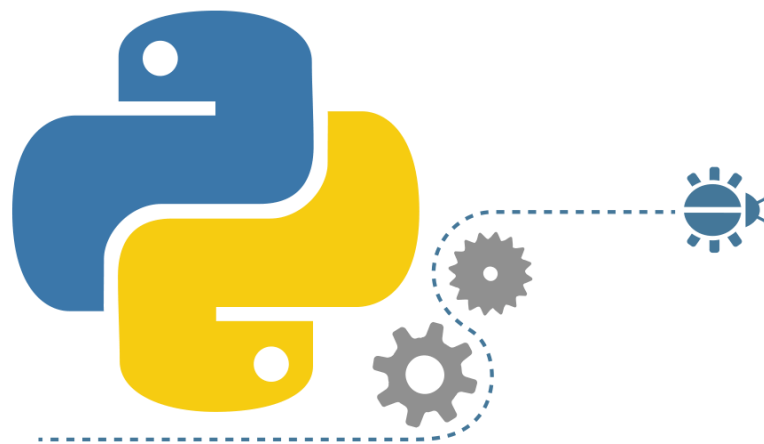
Chuyên đề ngôn ngữ lập trình

Chương 3



Cần Thơ, ngày 9 tháng 10 năm 2019

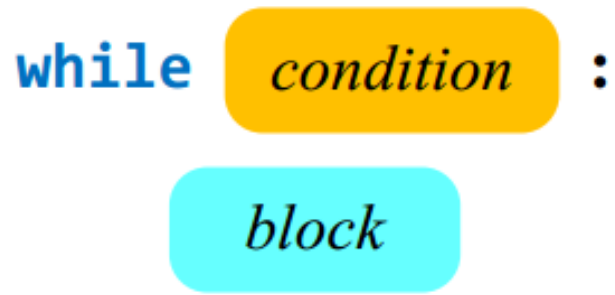
Vòng lặp while



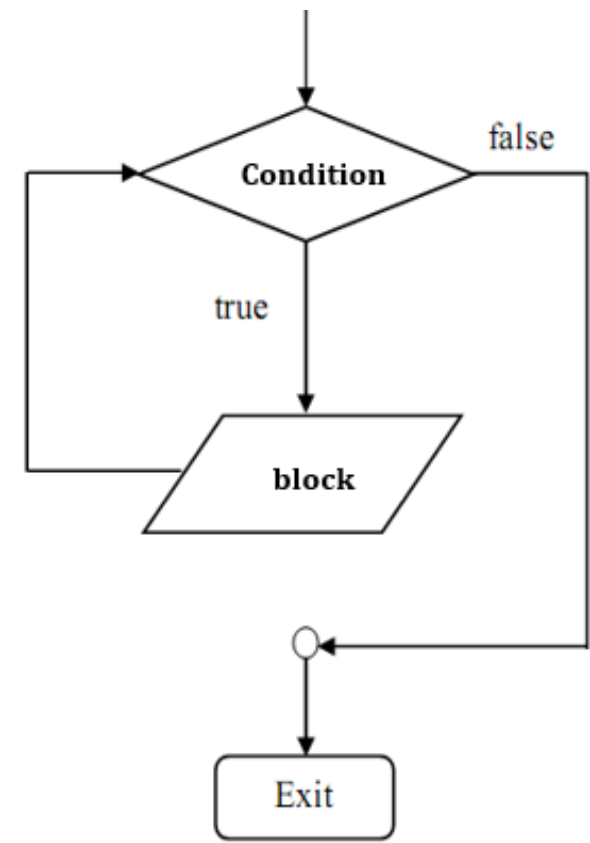
Nội dung bài học

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:



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

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

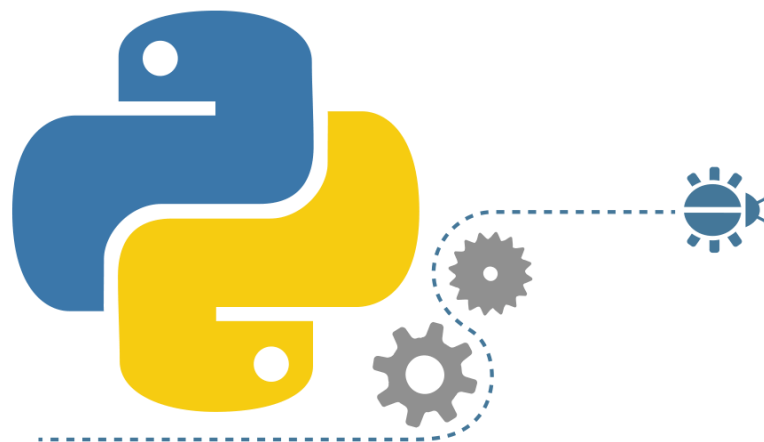
5) $i \leq n \Leftrightarrow 5 \leq 5 \rightarrow \text{True}$

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

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

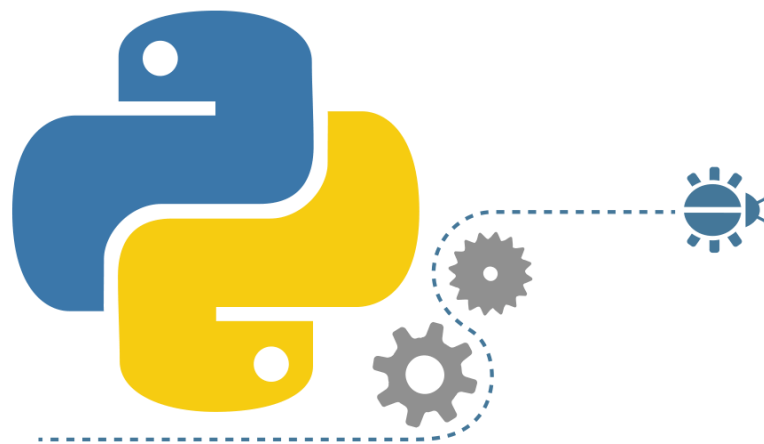
2) $x=4 \Rightarrow s=s+x=2+4=6$

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.

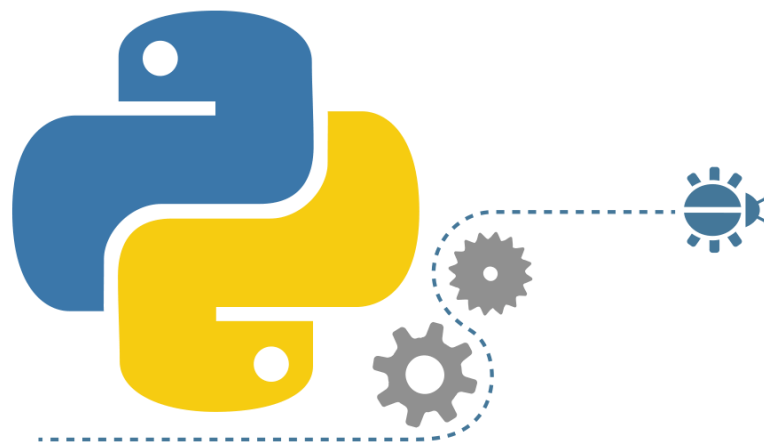
Gặp lệnh **break**, chương trình sẽ không thực hiện bất cứ lệnh nào bên dưới nó, mà thoát luôn khỏi vòng lặp.

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.

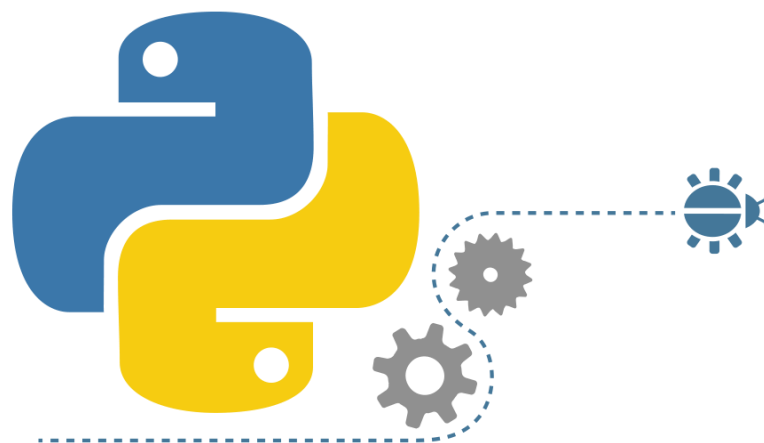
Nội dung bài học

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)
```

Lệnh while...else



Nội dung bài học

Python hỗ trợ **else block** trong trường hợp **while** kết thúc một cách bình thường (tức là không phải dùng **break** để kết thúc)

```
while condition:  
    while-block  
else:  
    else-block
```

Nếu **while** kết thúc một cách bình thường thì **else-block** sẽ tự động được thực hiện ngay sau đó.

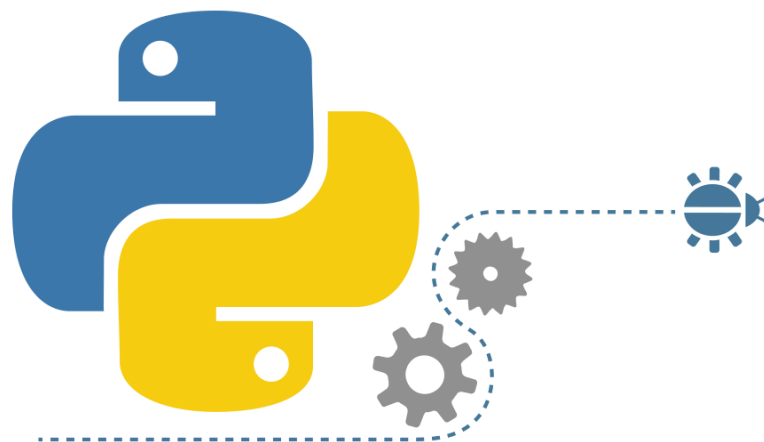
Nội dung bài học

Ví dụ:

```
count = sum = 0
print('Nhập danh sách các số dương để tính trung')
while count < 5:
    val = float(input('Nhập số: '))
    if val < 0:
        print('Số 0 sai quy tắc, thoát phần mềm')
        break
    count += 1
    sum += val
else:
    print('Trung Bình =', sum/count)
```

Nếu nhập số **<0** thì sẽ thoát **while** và bên trong **else** cũng không được thực thi (do kết thúc bằng lệnh **break**). Nếu nhập đúng toàn bộ giá trị, sau khi **while** chạy xong thì **else** sẽ tự động được gọi (kết thúc **while** bình thường)

Lệnh for...else



Nội dung bài học

Python hỗ trợ **else block** trong trường hợp **for** kết thúc một cách bình thường (tức là không phải dùng **break** để kết thúc)

```
for expression:  
    for-block  
else:  
    else-block
```

Nếu **for** kết thúc một cách bình thường thì **else-block** sẽ tự động được thực hiện ngay sau đó.

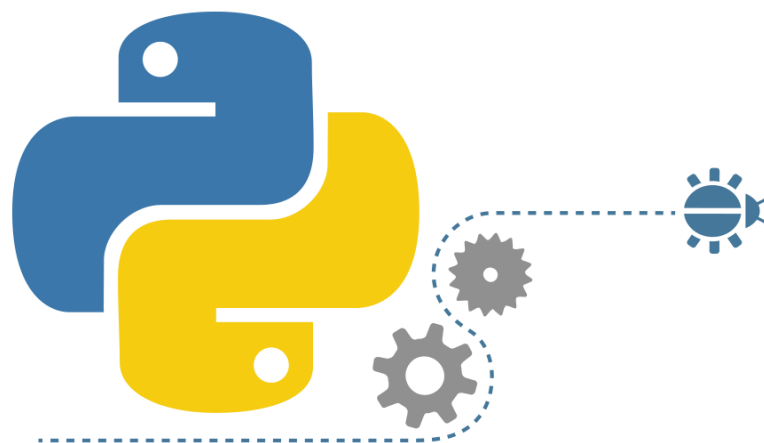
Nội dung bài học

Ví dụ:

```
a=int(input("Nhập a:"))
s=0
for n in range(5,10):
    if 4%a is 1:
        print("Ngừng for")
        break
    s=s+n
else:
    print("Sum=",s)
```

Nếu nhập **a** là số chẵn thì tính ra **sum**, còn số lẻ không ra **sum** (do ngừng for bằng **break**)

Vòng lặp lồng nhau



Nội dung bài học

Python cũng như các ngôn ngữ khác, ta có thể viết các vòng lặp lồng nhau.

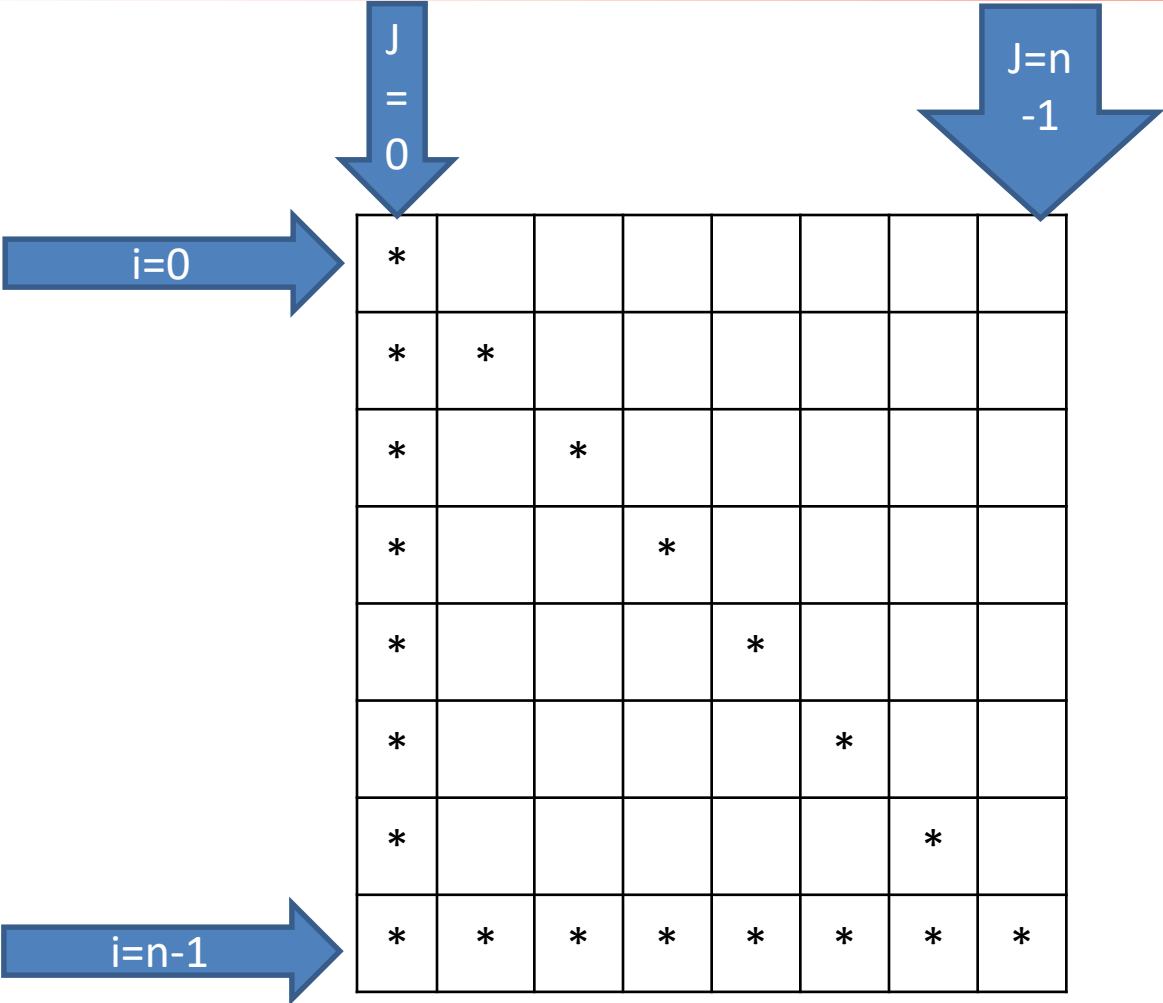
```
n=int(input("Nhập chiều cao:"))
for i in range(n):
    for j in range(n):
        if j==0 or i==j or j==n-1:
            print("*",end='')
        else:
            print(" ",end='')
    print()
```



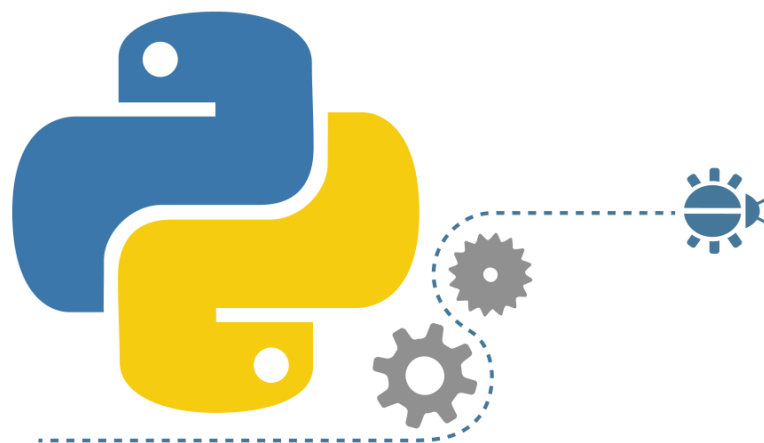
Nhập chiều cao:7

```

*      *
**     *
* *    *
*  *   *
*   *  *
*    **
*     *
```



Bài tập rèn luyện Tính dãy số

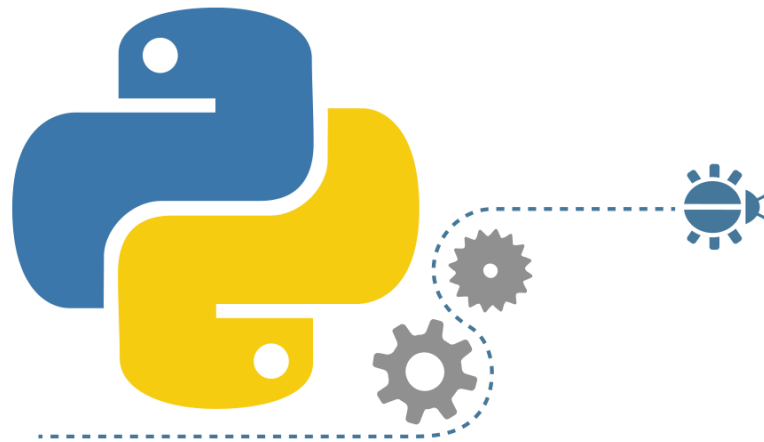


Nội dung bài học

$$\text{Tính } S(x, n) = x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

$$S(2,3)=2+2^2/2!+2^3/3!=2+4/2+8/6=2+2+4/3=5.3333$$

Bài tập rèn luyện Kiểm tra số nguyên tố

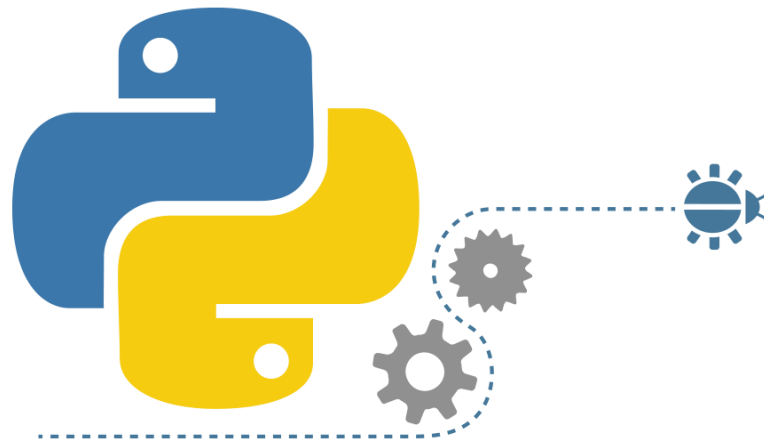


Nội dung bài học

Viết chương trình nhập vào một số, kiểm tra xem số này có phải là số nguyên tố hay không.

Hỏi người dùng có tiếp tục sử dụng hay thoát phần mềm.

Bài tập rèn luyện Xuất bản cứu chương



Nội dung bài học

Xuất bảng cửu chương 2→9 như hình dưới đây

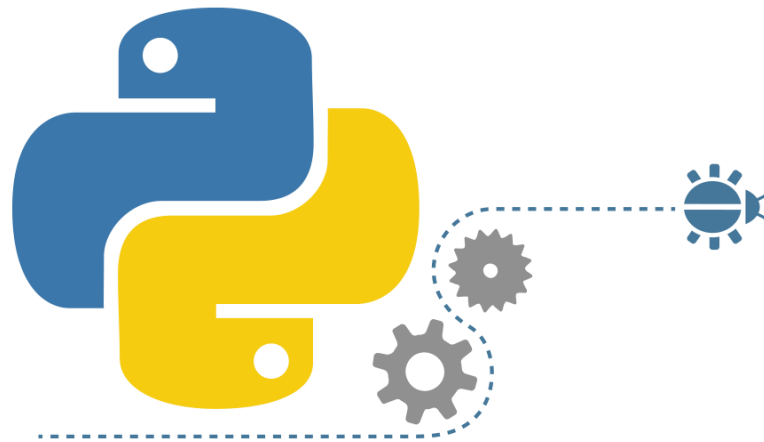
2* 1 = 2	3* 1 = 3	4* 1 = 4	5* 1 = 5	6* 1 = 6	7* 1 = 7	8* 1 = 8	9* 1 = 9
2* 2 = 4	3* 2 = 6	4* 2 = 8	5* 2 =10	6* 2 =12	7* 2 =14	8* 2 =16	9* 2 =18
2* 3 = 6	3* 3 = 9	4* 3 =12	5* 3 =15	6* 3 =18	7* 3 =21	8* 3 =24	9* 3 =27
2* 4 = 8	3* 4 =12	4* 4 =16	5* 4 =20	6* 4 =24	7* 4 =28	8* 4 =32	9* 4 =36
2* 5 =10	3* 5 =15	4* 5 =20	5* 5 =25	6* 5 =30	7* 5 =35	8* 5 =40	9* 5 =45
2* 6 =12	3* 6 =18	4* 6 =24	5* 6 =30	6* 6 =36	7* 6 =42	8* 6 =48	9* 6 =54
2* 7 =14	3* 7 =21	4* 7 =28	5* 7 =35	6* 7 =42	7* 7 =49	8* 7 =56	9* 7 =63
2* 8 =16	3* 8 =24	4* 8 =32	5* 8 =40	6* 8 =48	7* 8 =56	8* 8 =64	9* 8 =72
2* 9 =18	3* 9 =27	4* 9 =36	5* 9 =45	6* 9 =54	7* 9 =63	8* 9 =72	9* 9 =81
2*10 =20	3*10 =30	4*10 =40	5*10 =50	6*10 =60	7*10 =70	8*10 =80	9*10 =90

Nội dung bài học

Coding mẫu:

```
for i in range(1, 11):  
    for j in range(2, 10):  
        line = "{0}*{1:>2}={2:>2}".format(j, i, i * j)  
        print(line, end='\t')  
    print()
```

Các bài tập tự rèn luyện



Nội dung bài học

Câu 1: Cho biết bao nhiêu dấu * được in ra trên màn hình

```
a = 0
while a < 100:
    print('*', end='')
    print()
```

Câu 2: Cho biết bao nhiêu dấu * được in ra trên màn hình

```
a = 0
while a < 100:
    b = 0
    while b < 40:
        if (a + b) % 2 == 0:
            print('*', end='')
        b += 1
    print()
    a += 1
```

Nội dung bài học

Câu 3: Giải thích cách chạy các dòng lệnh range:

- (a) `range(5)`
- (b) `range(5, 10)`
- (c) `range(5, 20, 3)`
- (d) `range(20, 5, -1)`
- (e) `range(20, 5, -3)`
- (f) `range(10, 5)`
- (g) `range(0)`
- (h) `range(10, 101, 10)`
- (i) `range(10, -1, -1)`
- (j) `range(-3, 4)`
- (k) `range(0, 10, 1)`

Nội dung bài học

Câu 4: Bao nhiêu dấu * được in ra màn hình

```
for a in range(20, 100, 5):  
    print('*', end='')  
print()
```

Câu 5: Viết lại coding dưới đây bằng cách dùng từ khóa break thay thế cho biến done:

```
done = False  
n, m = 0, 100  
while not done and n != m:  
    n = int(input())  
    if n < 0:  
        done = True  
    print("n =", n)
```

Nội dung bài học

Câu 6: Vẽ các hình dưới đây

```

* * * *
*      *
*      *
*      *
* * * *

```

```

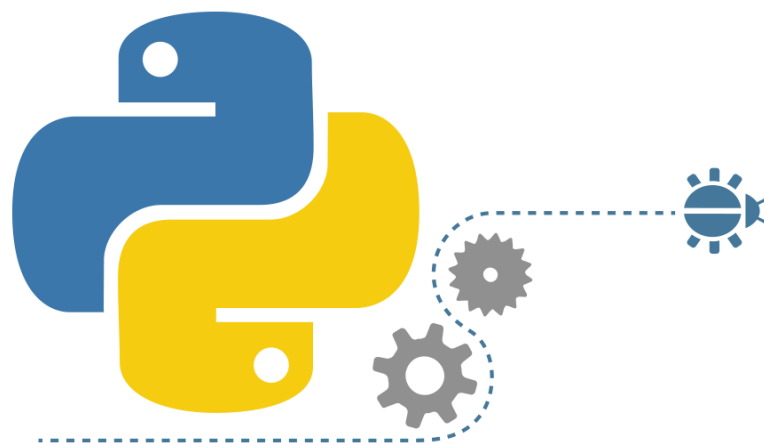
*
* *
*   *
* * * * *
*   *   *
*       *
*       *
*       *

```

Câu 7: Nhập x, n, tính S(x,n):

Nhập x,n. Tính $S(x,n) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + ... + \frac{x^{2n+1}}{(2n+1)!}$

Các hàm toán học



Nội dung bài học

Một số hàm ta phải sử dụng từ các thư viện có sẵn, như các hàm tính căn bậc 2, lũy thừa, log... Bài giảng này ta sẽ tập sử dụng:

- Sqrt-Căn bậc 2
- Pow – lũy thừa
- Log-> $\log(x)=\log_e x=\ln x$
- Log10- Logarit cơ số 10 của x, $\log_{10}(x)=\log_{10} x$
- Exp-tính e^x
- Degrees-Đổi radian ra độ
- Radians- Tính radian $180/PI*x$
- Fabs- tính giá trị tuyệt đối

Nội dung bài học

Ví dụ:

```
from math import *  
print('sqrt(25)=', sqrt(25))  
print('pow(5,3)=', pow(5,3))  
print('log(2)=', log(2))  
print('log10(100)=', log10(100))  
print('exp(2)=', exp(2))  
print(degrees(0.5235987755982988))  
print(radians(30))
```



```
sqrt(25)= 5.0  
pow(5,3)= 125.0  
log(2)= 0.6931471805599453  
log10(100)= 2.0  
exp(2)= 7.38905609893065  
29.999999999999996  
0.5235987755982988
```

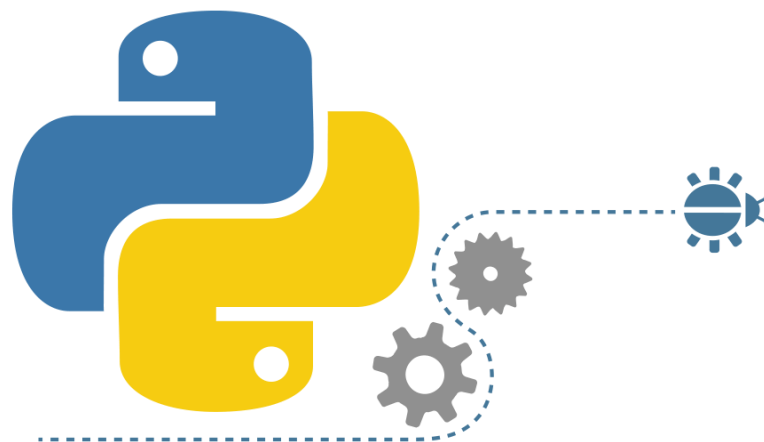
Nội dung bài học

Bài học này trình bày cách sử dụng các hàm Lượng giác trong Python

-Từ một góc ta đưa về Radian sau đó đẩy vào các hàm lượng giác:

- Sin
- Cos
- tan

Hàm round



Nội dung bài học

Trong quá trình tính toán ta sẽ nhận được các số lẻ, các số này cần được làm tròn để cùng thỏa mãn yêu cầu nào đó. Hàm round được Python hỗ trợ như sau:

- `round(số gốc, đơn vị làm tròn)`

```
a=3
```

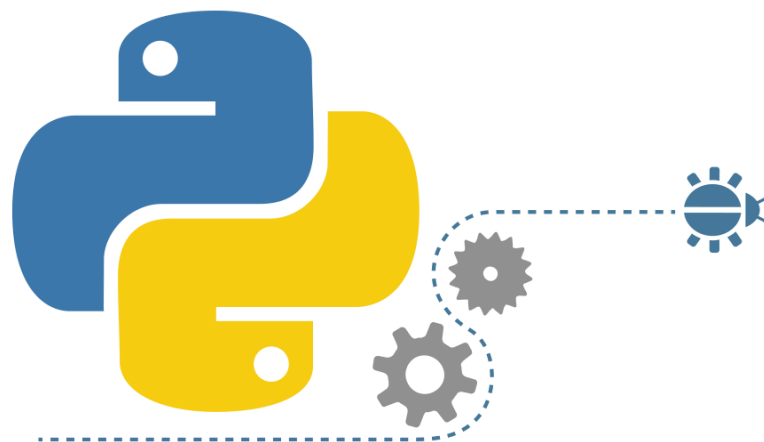
```
b=11
```

```
c=b/a
```

```
print(c) → 3.666666666666666665
```

```
print(round(c, 2)) → 3.67
```

Hàm Time



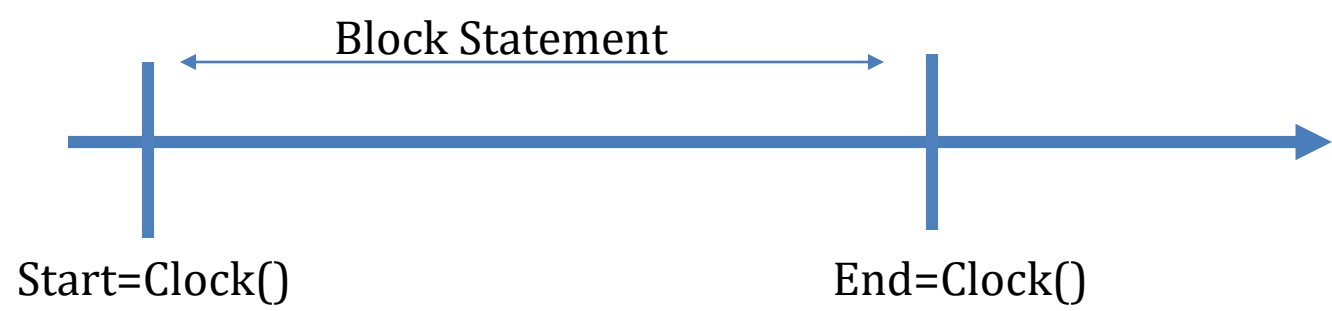
Nội dung bài học

Module time trong Python cung cấp rất nhiều hàm hữu ích, bài học giới thiệu 2 hàm **clock** và **sleep**.

1. Clock
2. Sleep

1. Clock

Clock tùy thuộc vào hệ điều hành mà cách thức xử lý khác nhau. Với **Windows** thì **Clock** trả về **số giây** khi ta gọi hàm **clock**. Để tính thời gian thực hiện một chương trình ta có thể căn 2 đầu để trừ ra số giây



Thời gian thực thi=End-Start



1. Clock

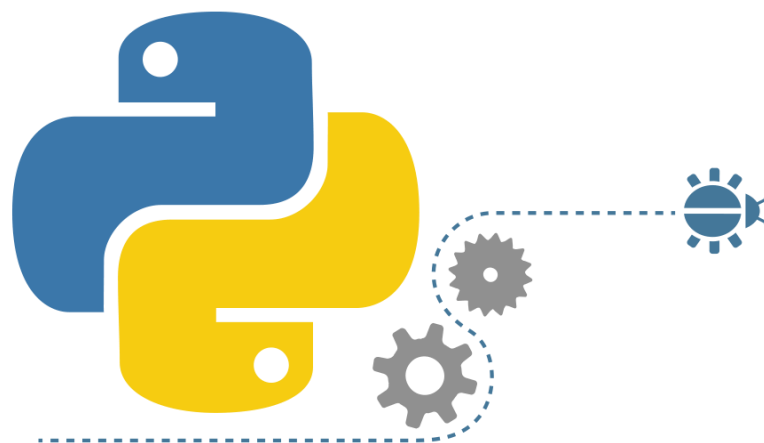
```
from time import clock
print("Enter your name: ", end="")
start_time = clock()
name = input()
elapsed = clock() - start_time
print(name, "it took you", elapsed, "seconds to respond")
```

2. Sleep

Sleep giúp ta tạm dừng quá trình chạy trong một đơn vị thời gian nào đó. Thay vì chương trình chạy một mạch, khi gặp sleep nó sẽ tạm dừng lại

```
from time import sleep
for count in range(10, -1, -1): # Range 10, 9, 8, ..., 0
    print(count) # Display the count
    sleep(1) # Suspend execution for 1 second
```

Hàm Random

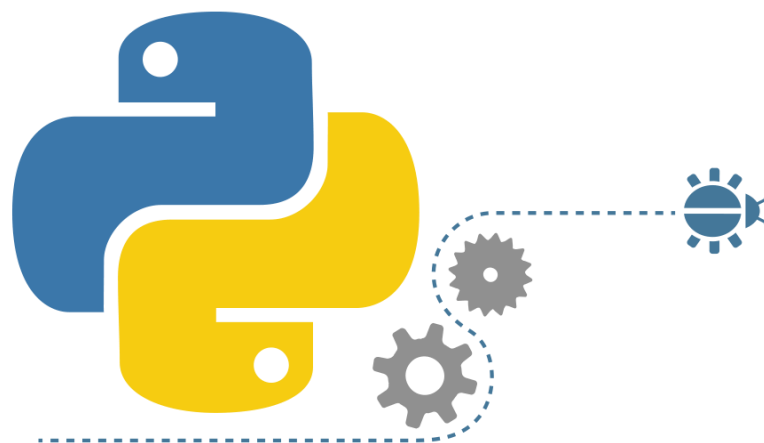


Nội dung bài học

Random là một trong những hàm khá hữu dụng trong việc viết Games, giả lập dữ liệu, thống kê.

`randrange(x,y)` → lấy số ngẫu nhiên $\geq x$ và $< y$

Hàm Exit

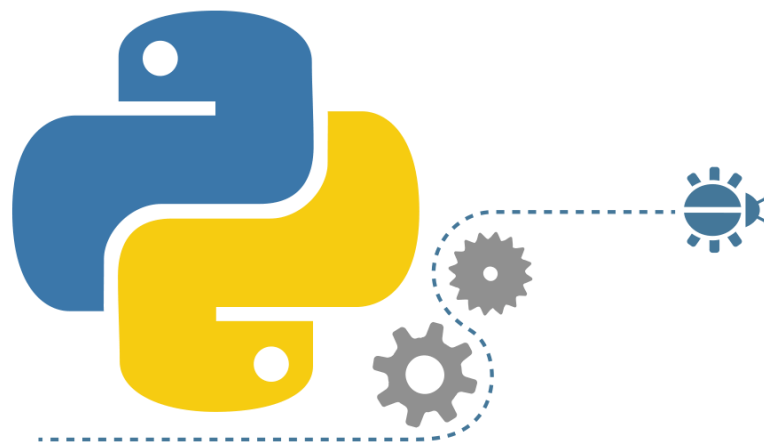


Nội dung bài học

Hàm exit dùng để thoát phần mềm

```
while True:
    s=input("Tên bạn:")
    print(s)
    hoi=input("Tiếp hay không? (c/k) :")
    if hoi=="k":
        exit()
print("BYE!")
```

Hàm Eval



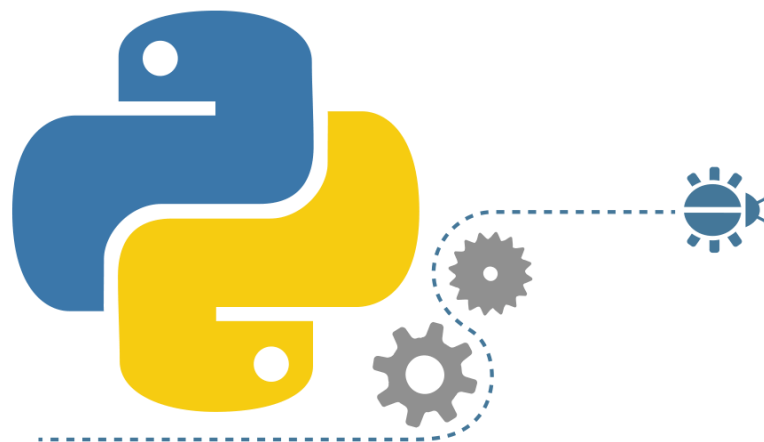
Nội dung bài học

Hàm eval rất lợi hại, nó có thể tự tính toán chuỗi phép toán:

```
from math import sin
x=eval("1+2+5+sin(30)")
print(x)
```

```
x1,x2=eval(input("Nhập x1,x2:"))
print("x1=",x1,"",x2=" ",x2)
print("{0}+{1}={2}".format(x1,x2,x1+x2))
```

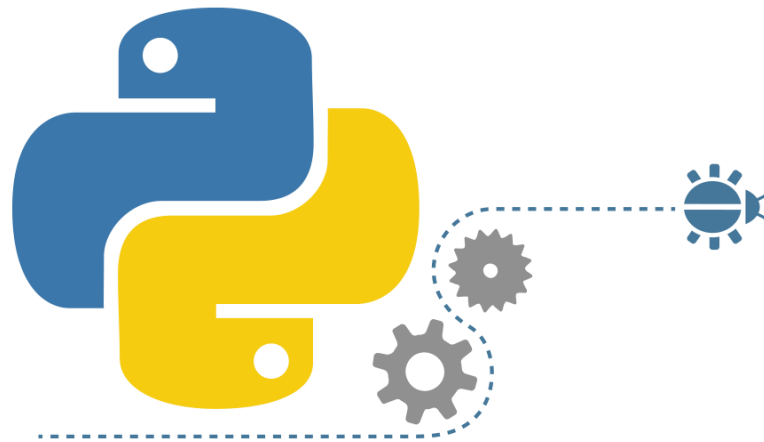
Bài tập rèn luyện Game đoán số



Nội dung bài học

- ❖ Máy ra 1 số trong đoạn $[1...100]$
- ❖ Người chơi đoán số, chỉ được phép đoán sai 7 lần. Mỗi lần đoán sẽ thông báo số người chơi đoán nhỏ hơn hay lớn hơn số của máy và hiển thị số lần đoán
- ❖ Game kết thúc khi: Đoán sai quá 7 lần hoặc đoán trúng trước 7 lần.
- ❖ Sau khi game kết thúc hỏi người chơi có tiếp tục hay không?

Bài tập rèn luyện Tính diện tích tam giác

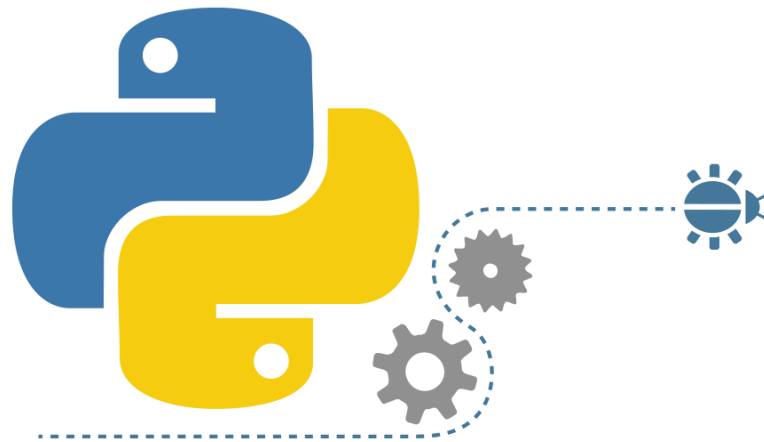


Nội dung bài học

Nhập vào 3 cạnh của tam giác, kiểm tra tính hợp lệ của tam giác
Sau đó tính diện tích theo công thức Herong:

$$cv=a+b+c, p=cv/2 \text{ và } dt=\sqrt{p*(p-a)*(p-b)*(p-c)}$$

Bài tập rèn luyện Tính điểm trung bình



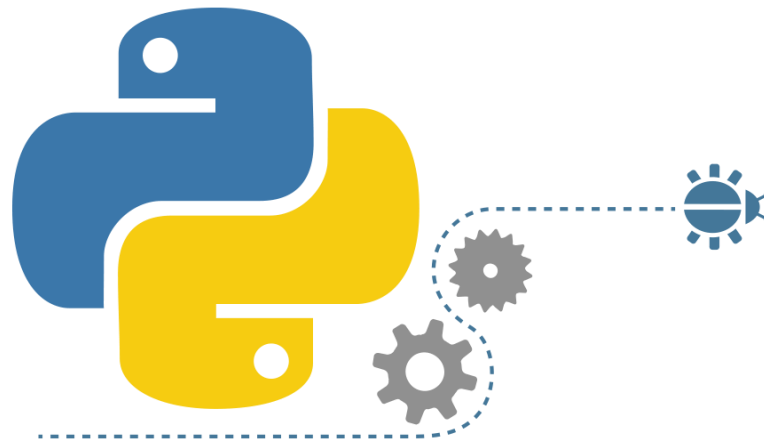
Nội dung bài học

Nhập vào điểm toán lý hóa bằng chuỗi, theo thứ tự:

Toán, lý, hóa → “7,4,6”

Tính điểm trung bình lấy 2 chữ số lẻ thập phân.

Các bài tập tự rèn luyện



Nội dung bài học

Câu 1: Những giá trị có thể xuất hiện khi chạy randrange(0, 100)

4.5 , 34 , -1, 100, 0, 99

Câu 2: Nhập toạ độ 2 điểm $A(x_A, y_A)$, $B(x_B, y_B)$. Tính và xuất độ dài đoạn AB.

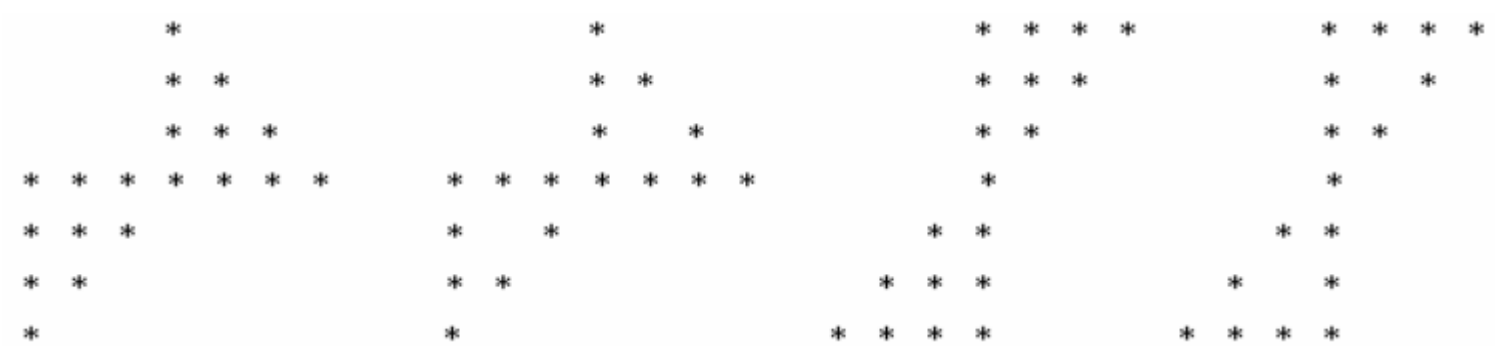
$$|AB|=d_{AB}=\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Câu 3: Viết chương trình tính $\log_a x$ với a, x là các số thực nhập vào từ bàn phím, và $x>0$, $a>0$, $a \neq 1$.(dùng $\log ax=\ln x/\ln a$)

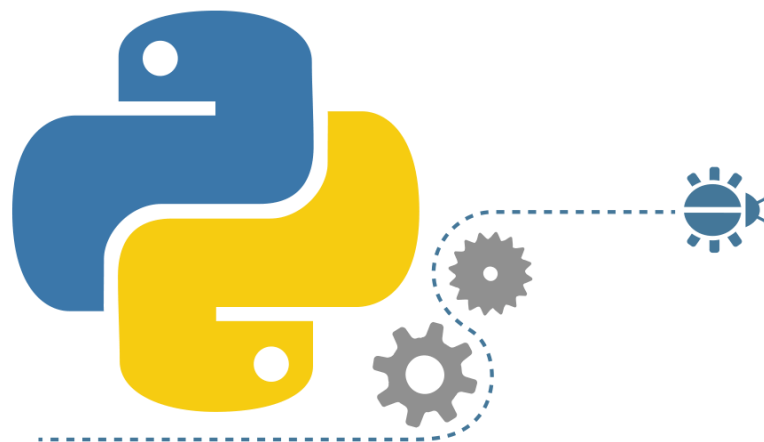
Nội dung bài học

Câu 4: Nhập n. Tính $S(n) = \sqrt{2 + \sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}}$, có n dấu căn lồng nhau.

Câu 5: Vẽ 4 hình dưới đây, dùng sleep để xuất hiện từng hình sau 5 giây



Khái niệm về Hàm



Nội dung bài học

- Hàm là một khối lệnh thực hiện một công việc hoàn chỉnh (module), được đặt tên và được gọi thực thi nhiều lần tại nhiều vị trí trong chương trình.
- Hàm còn gọi là chương trình con (*subroutine*)
- ***Nếu không viết hàm thì sẽ gặp những khó khăn gì?***
 - Rất khó để viết chính xác khi dự án lớn
 - Rất khó debug
 - Rất khó mở rộng

Nội dung bài học

- Có hai loại hàm:
 - *Hàm thư viện*: là những hàm đã được xây dựng sẵn. Muốn sử dụng các hàm thư viện phải khai báo thư viện chứa nó trong phần khai báo `from ... import`.
 - *Hàm do người dùng định nghĩa*.

Nội dung bài học

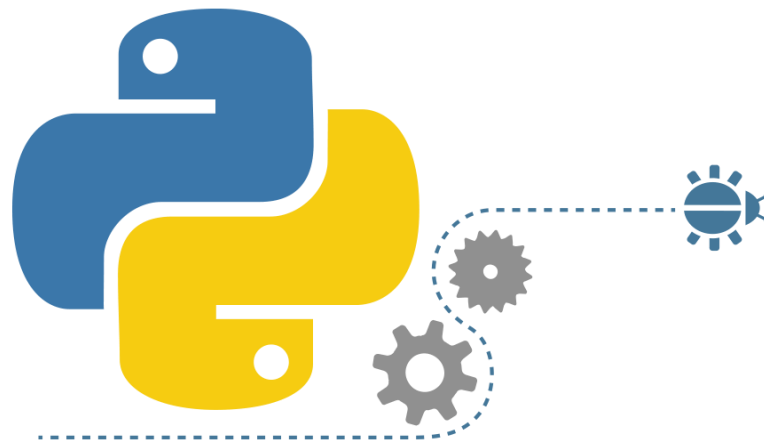
- Ví dụ hàm thư viện:

```
print("Chương trình tính điểm trung bình")
toan, ly, hoa = eval(input("Nhập điểm toán, lý, hóa: "))
print("Điểm toán=", toan)
print("Điểm lý=", ly)
print("Điểm hóa=", hoa)
dtb = (toan + ly + hoa) / 3
print("Điểm trung bình=", dtb)
print("Điểm làm tròn=", round(dtb, 2))
```

- Ví dụ hàm tự định nghĩa:

```
def cong(x, y):
    return x + y
```

Cấu trúc tổng quát của hàm



Nội dung bài học

- Python có cấu trúc tổng quát khi khai báo Hàm như sau:

```
def name ( parameter list ) :  
    block
```

- Dùng từ khóa **def** để định nghĩa hàm, các hàm có thể có đối số hoặc không. Có thể trả về kết quả hoặc không

Nội dung bài học

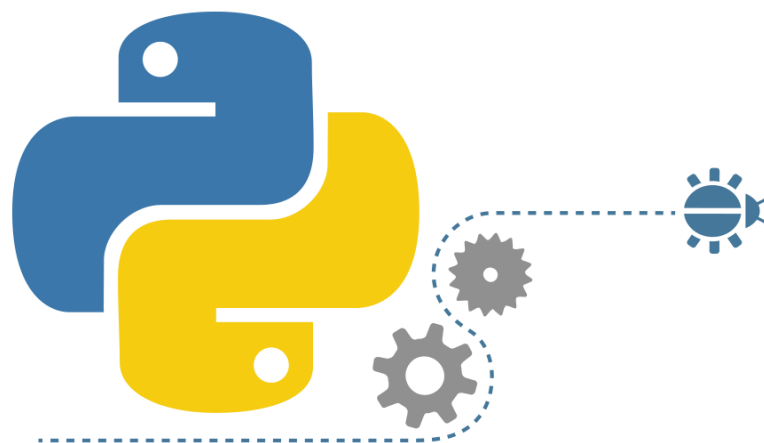
- Ví dụ : Viết hàm tính giải phương trình bậc 1

```
1 def PTB1(a,b):  
2     if a ==0 and b==0:  
3         return "Vô số nghiệm"  
4     elif a==0 and b!=0:  
5         return "Vô nghiệm"  
6     else:  
7         return "x={0}".format(round(-b/a,2))
```

- Ví dụ: Viết hàm xuất dữ liệu ra màn hình

```
def XuatDuLieu(data):  
    print(data)
```

Cách gọi hàm



Nội dung bài học

- Để gọi hàm ta cũng cần phải kiểm tra Hàm đó được định nghĩa như thế nào?
 - ✓ Có đối số hay không?
 - ✓ Có trả về kết quả hay không?

Nếu có kết quả trả về:

Result=FunctionName ([parameter])

Nếu không có kết quả trả về:

FunctionName([parameter])

Nội dung bài học

- Ở bài học trước ta có ví dụ về phương trình bậc 1 và xuất dữ liệu

```
def PTB1(a,b):  
    if a ==0 and b==0:  
        return "Vô số nghiệm"  
    elif a==0 and b!=0:  
        return "Vô nghiệm"  
    else:  
        return "x={0}".format(round(-b/a,2))
```

- Hàm PTB1 vừa có đối số vừa có kết quả trả về. Ta có thể gọi như sau:
kq=PTB1(5,8)

Nội dung bài học

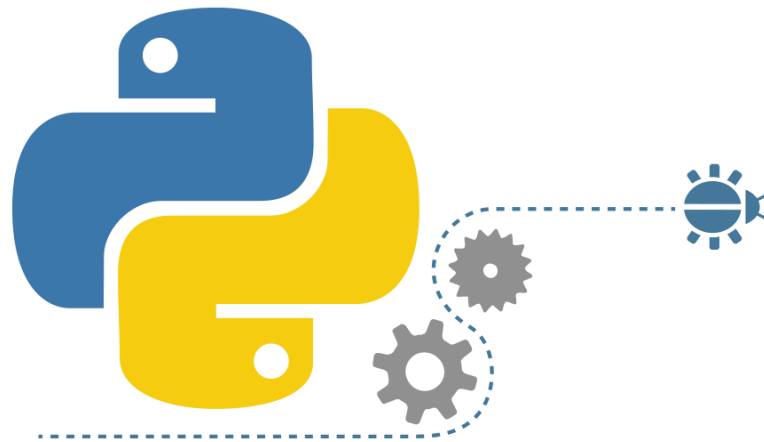
➤ Hàm xuất dữ liệu

```
def XuatDuLieu(data):  
    print(data)
```

Ta có thể gọi:

XuatDuLieu("hello")

Nguyên tắc hoạt động của hàm



Nội dung bài học

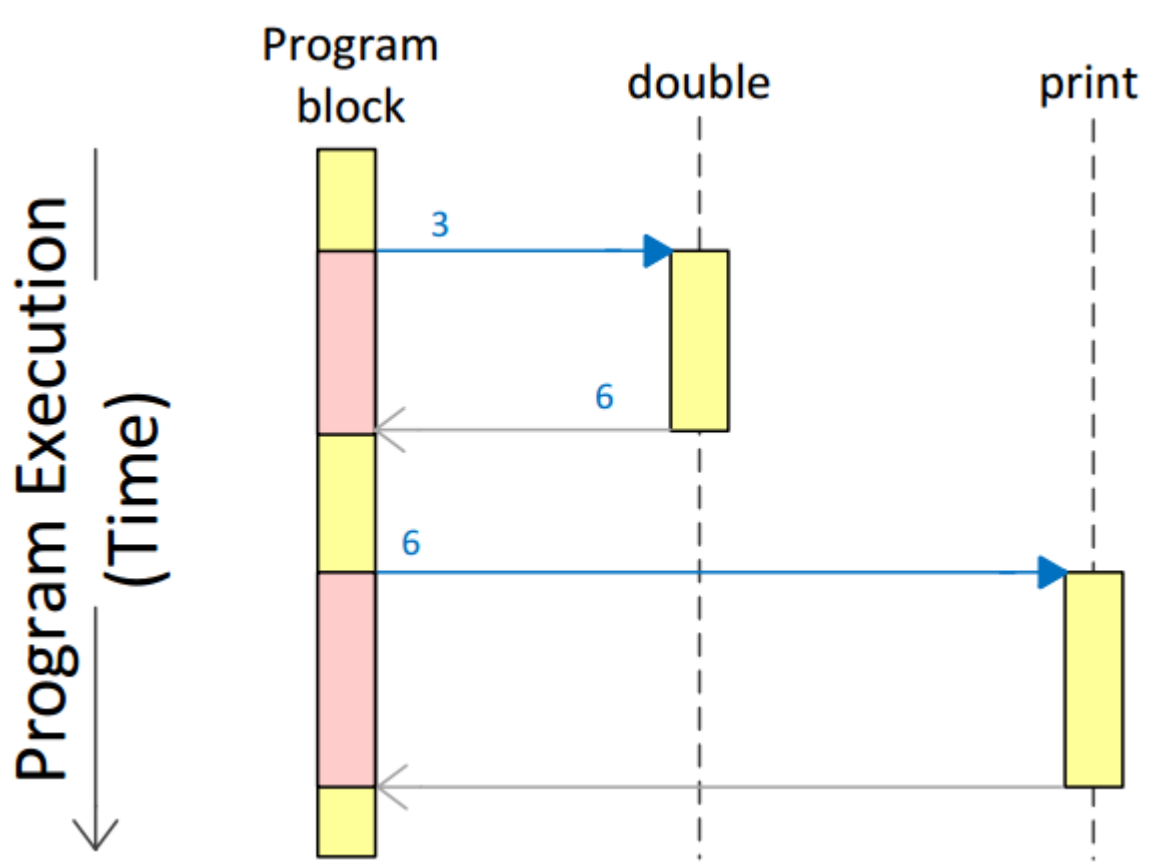
- Hàm trong Python cũng như trong các ngôn ngữ lập trình khác, đều hoạt động theo Nguyên tắc LIFO (LAST IN FIRST OUT)
- Ví dụ ta có các mã lệnh dưới đây:

```
def double(n):  
    return 2 * n  
  
x = double(3)  
print(x)
```

- Theo LIFO thì nó xảy ra như thế nào?

Nội dung bài học

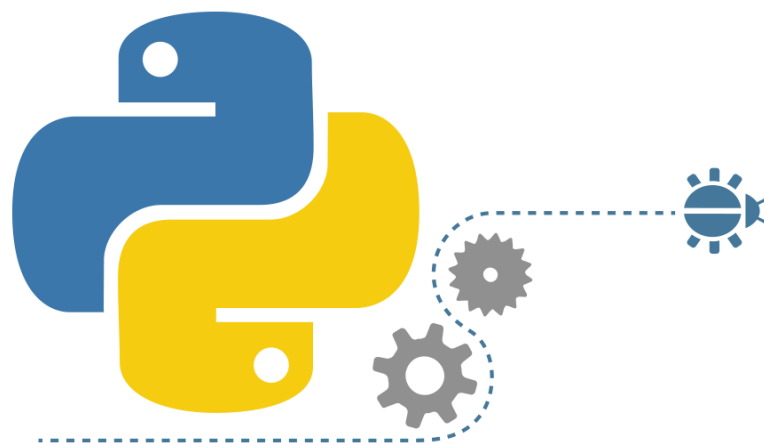
➤ Minh họa nguyên tắc hoạt động khi gọi hàm



```
def double(n):
    return 2 * n

x = double(3)
print(x)
```

Viết tài liệu cho hàm



Nội dung bài học

- Python hỗ trợ ta bổ sung tài liệu cho hàm, việc này rất thuận lợi cho đối tác sử dụng các function do ta làm ra. Dựa vào những tài liệu này mà Partner có thể dễ dàng biết cách sử dụng.
- Ta có thể sử dụng 3 dấu nháy kép hoặc 3 dấu nháy đơn để viết tài liệu cho hàm. Tuy nhiên theo kinh nghiệm thì các bạn nên dùng 3 dấu nháy kép.
- Các ghi chú (tài liệu) phải được viết ở những dòng đầu tiên khi khai báo hàm

Nội dung bài học

Ví dụ ta tạo 1 file

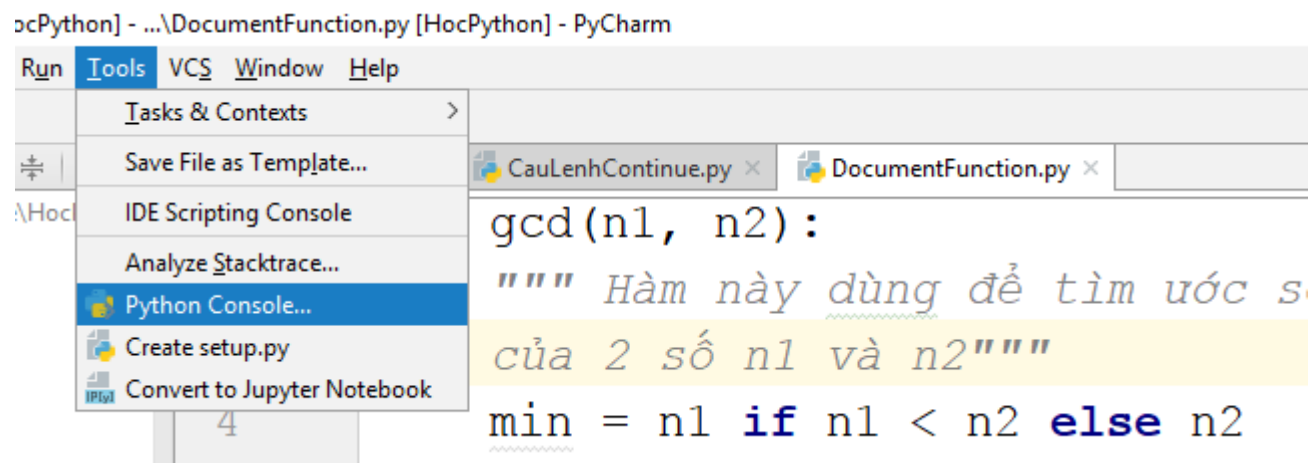
DocumentFunction.py

Có 2 hàm **gcd** và **ptb1**

```
1 def gcd(n1, n2):
2     """ Hàm này dùng để tìm ước số chung lớn nhất
3     của 2 số n1 và n2 """
4     min = n1 if n1 < n2 else n2
5     largest_factor = 1
6     for i in range(1, min + 1):
7         if n1 % i == 0 and n2 % i == 0:
8             largest_factor = i
9     return largest_factor
10 def ptb1(a,b):
11     """Giải phương trình bậc 1
12     ax+b=0"""
13     if a ==0 and b==0:
14         return "Vô số nghiệm"
15     elif a==0 and b!=0:
16         return "Vô nghiệm"
17     else:
18         return "x={0}".format(round(-b/a,2))
```

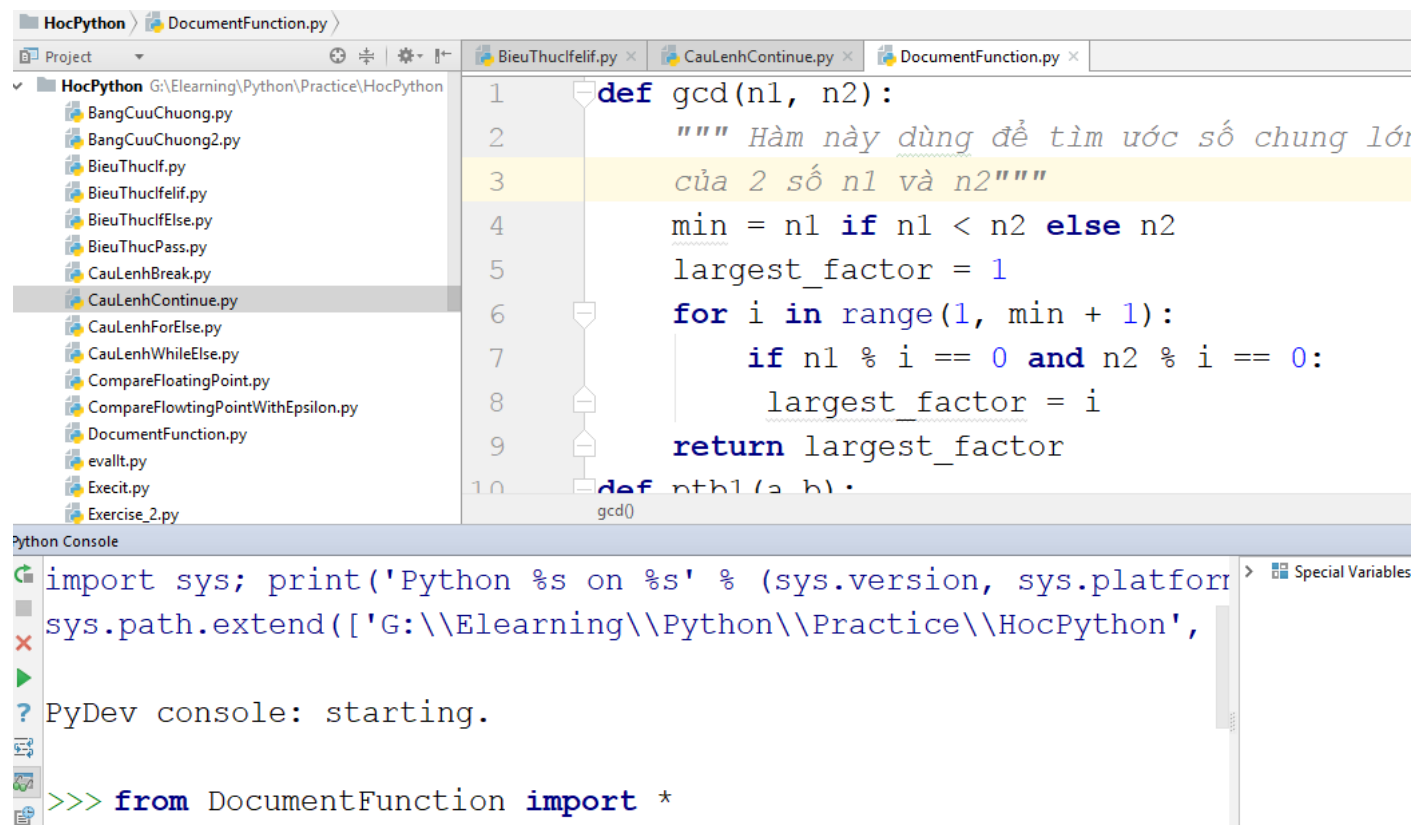
Nội dung bài học

Ta chạy command line để xem cách lấy tài liệu cho các hàm trên. Ta vào menu tools/chọn Python Console...



Nội dung bài học

Ta chạy command line để xem cách lấy tài liệu cho các hàm trên. Ta vào menu tools/chọn Python Console...



The screenshot shows a Python IDE with a project named 'HocPython' located at 'G:\Elearning\Python\Practice\HocPython'. The project contains several Python files, including 'BangCuuChuong.py', 'BangCuuChuong2.py', 'BieuThucIf.py', 'BieuThucIfelif.py', 'BieuThucIfelse.py', 'BieuThucPass.py', 'CauLenhBreak.py', 'CauLenhContinue.py', 'CauLenhForElse.py', 'CauLenhWhileElse.py', 'CompareFloatingPoint.py', 'CompareFlowtingPointWithEpsilon.py', 'DocumentFunction.py', 'evalIt.py', 'Execit.py', and 'Exercise_2.py'. The 'DocumentFunction.py' file is open in the editor, showing a function 'gcd(n1, n2)' that calculates the greatest common divisor (GCD) of two numbers. The function uses a loop to find the largest factor that divides both numbers. The Python Console window at the bottom shows the following code being executed:

```
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['G:\\Elearning\\Python\\Practice\\HocPython',
PyDev console: starting.
>>> from DocumentFunction import *
```

from DocumentFunction import *

Nội dung bài học

Muốn xem tài liệu của hàm nào thì gõ : **help**(function name)

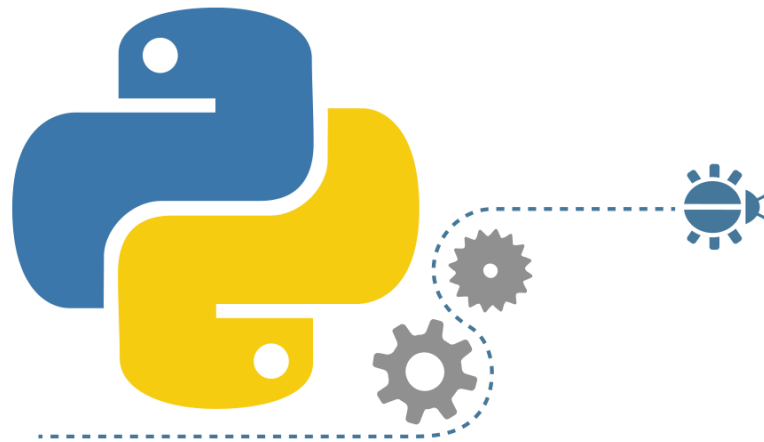
```
>>> from DocumentFunction import *
>>> help(gcd)
Help on function gcd in module DocumentFunction:

gcd(n1, n2)
    Hàm này dùng để tìm ước số chung lớn nhất
    của 2 số n1 và n2
```

```
>>> help(ptb1)
Help on function ptb1 in module DocumentFunction:

ptb1(a, b)
    Giải phương trình bậc 1
    ax+b=0
```

Global Variable



Nội dung bài học

- Tất cả các biến khai báo trong hàm chỉ có phạm vi ảnh hưởng trong hàm, các biến này gọi là biến local. Khi thoát khỏi hàm thì các biến này không thể truy xuất được.

- Xem ví dụ sau:

```
1 g=5 → Global variable
2 def increment():
3     g=2 → Local variable
4     g=g+1
5     increment()
6     print(g) → Global variable
```

- Theo bạn thì chạy xong 6 dòng lệnh ở trên, giá trị g xuất ra màn hình bao nhiêu?

Nội dung bài học

➤ Xem ví dụ 2 sau:

```
1 g=5
2 def increment():
3     global g
4     g=2
5     g=g+1
6     increment()
7     print(g)
```

- Theo bạn thì chạy xong 7 dòng lệnh ở trên, giá trị g xuất ra màn hình bao nhiêu?
- Global cho phép ta tham chiếu sử dụng được biến Global

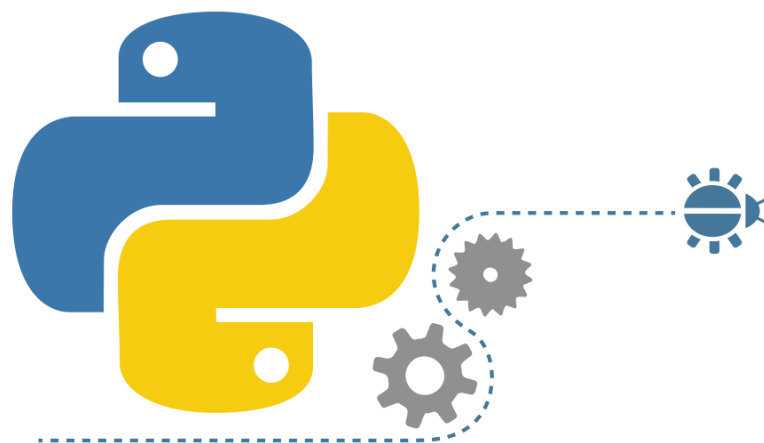
Nội dung bài học

➤ Xem ví dụ 3 sau:

```
1 g=5
2 def increment():
3     g=g+1
4     increment()
5     print(g)
```

➤ G dòng 3 Báo lỗi nha, vì g ở trong hàm không có lấy g ở ngoài (khai báo ở dòng 1)

Parameter mặc định



Nội dung bài học

- Python cũng tương tự như C++, có hỗ trợ Parameter mặc định khi Khai báo hàm.
- Hàm print ta sử dụng cũng có các parameter mặc định

```
* def print(self, *args, sep=' ', end='\n', file=None): #  
    """  
    print(value, ..., sep=' ', end='\n', file=sys.stdout
```


Nội dung bài học

- Vậy nếu tự viết hàm thì ta sẽ định nghĩa các Parameter mặc định này như thế nào?

```
1  def SumRange (n, m=0) :  
2      sum=0  
3      for i in range (1, m+n, 1) :  
4          sum=i  
5      return sum  
6  
7  x1=SumRange (5)  
8  print ("x1=", x1)  
9  x2=SumRange (5, 1)  
10 print ("x2=", x2)
```

```
↓  
x1= 4  
x2= 5
```

END

