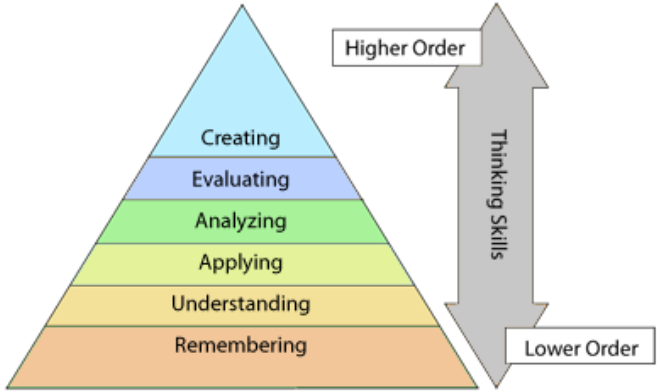
CẤU TRÚC LẶP (LOOP)

Kỹ Thuật Lập Trình Python

Giảng viên: ThS. Nghi Hoàng Khoa | Email: khoanh@uit.edu.vn

Mục tiêu

- Phân tích sử dụng đúng lệnh tạo vòng lặp
- Vận dụng lệnh tạo vòng lặp



Outline

1. Vòng lặp for
2. Vòng lặp while
3. Lệnh continue
4. Lệnh break
5. Lệnh pass

for

- ***Cú pháp***

```
for <biến vòng lặp> in <dãy giá trị> :
      <Lệnh hoặc khối lệnh>
```

- <dãy giá trị> thể hiện cho số lần lặp, bao gồm chuỗi, danh sách, tuple, dictionary, range…

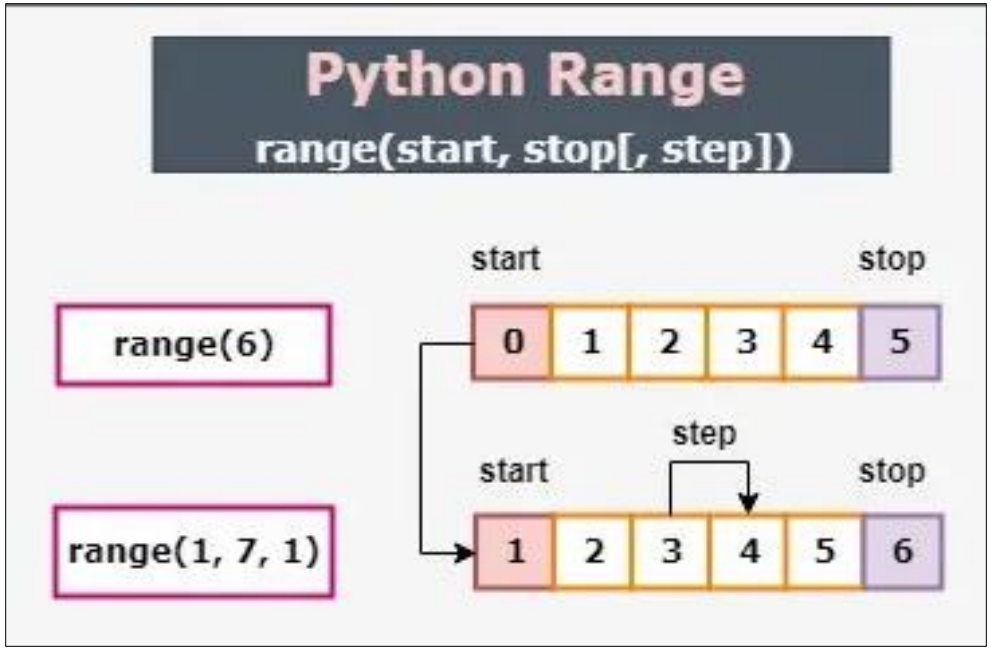
- ***Ví dụ minh họa***
  - Đếm số ký tự của chuỗi các phần tử trong danh sách cho trước.

```
colors =["Đỏ", "Cam", "Vàng", "Lục", "Lam", "Chàm", "Tím"]
for i in colors:
    print(i, " \t", len(i))
```

Kết quả:

```
Đỏ      Số ký tự: 2
Cam     Số ký tự: 3
Vàng    Số ký tự: 4
Lục     Số ký tự: 3
Lam     Số ký tự: 3
Chàm    Số ký tự: 4
Tím     Số ký tự: 3
```

- ***Nhắc lại***


Python Range
range(start, stop[, step])
start
stop
range(6)
0
1
2
3
4
5
step
start
stop
range(1, 7, 1)
1
2
3
4
5
6

for

- ***Áp dụng. Cách tạo vòng lặp từ khi biết trước số lần lặp (n lần):***

```
for i in range(n):
      <Lệnh hoặc khối lệnh>
      print(i)
```

```
for i in range(start, stop, step):
      <Lệnh hoặc khối lệnh>
      print(i)
```

for

- ***Ví dụ minh họa***

```
for i in range(10):
    print(i, end='\t')
```

```
0	1	2	3	4	5	6	7	8	9
```

```
for i in range(1, 10, 2):
    print(i, end='\t')
```

```
1	3	5	7	9
```

Ứng dụng for để tạo list mới

- Tạo một danh sách bằng cách kết hợp lệnh for và biểu thức

Cú pháp – còn gọi là cú pháp comprehension

new_list = **[**expression **for** item **in** iterable **if** condition**]**

- expression: biểu thức để tính toán giá trị của từng phần tử trong list mới.
- item: biến được sử dụng để lặp qua từng phần tử trong list ban đầu.
- condition: điều kiện kiểm tra mà các item phải thỏa để được thêm vào new_list

```
ds = [2, 4, 6, 8, 10]
new_ds = [item - 1 for item in ds]
print(new_ds)
```

- **BÀI #1**

Viết chương trình tính giai thừa của một số cho trước n (nguyên dương) được nhập từ bàn phím.

$$S = n! = 1.2.3...n$$

- **BÀI #2**
- Viết chương trình in ra bản cửu chương 2 đến cửu chương 9. Ví dụ:

1 x 2 = 2
2 x 2 = 4
3 x 2 = 6
4 x 2 = 8
5 x 2 = 10
6 x 2 = 12
7 x 2 = 14
8 x 2 = 16
9 x 2 = 18
10 x 2 = 20

- **BÀI #3**

Viết chương trình có thể tạo danh sách:

- Số lượng các phần tử được chọn ngẫu nhiên từ 20 đến 30.
- Các giá trị bình phương của các số thực được chọn ngẫu nhiên từ 18 đến 99.

while

- ***Cú pháp lệnh while***

```
while   <điều kiện kiểm tra>:
        <Lệnh hoặc khối lệnh>
```

- Cách nhớ:

  “kiểm tra điều kiện trước rồi mới thực hiện lệnh”

- ***Ví dụ minh họa***
  - Nhập vào một số dương, nếu không đúng bắt người dùng nhập lại đến khi nào đúng thì thôi (tức là dừng nhập).

```
x = float(input("Nhập x = "))

while x < 0:
    x = float(input("Nhập lại x = "))

print("Giá trị đã nhập:", x)
```

while

- ***Ví dụ minh họa***

```
count = 0
n = int(input("Nhập vào số lần cần lặp: "))
while (count < n):
    print ("Lần lập thứ:", count + 1,"\tBiến đếm:", count)
    count = count + 1
```

Kết quả:

```
Nhập vào số lần cần lặp: 8
Lần lập thứ: 1      Biến đếm: 0
Lần lập thứ: 2      Biến đếm: 1
Lần lập thứ: 3      Biến đếm: 2
Lần lập thứ: 4      Biến đếm: 3
Lần lập thứ: 5      Biến đếm: 4
Lần lập thứ: 6      Biến đếm: 5
Lần lập thứ: 7      Biến đếm: 6
Lần lập thứ: 8      Biến đếm: 7
```

BÀI #1

- Viết lệnh kiểm tra giá trị nhập vào từ bàn phím thỏa điều kiện thuộc [-99; 99]. Nếu không khỏa bắt người dùng nhập lại đến khi nào khỏa điều kiện.

BÀI #2

- Viết lệnh kiểm tra giá trị nhập vào từ bàn phím thỏa điều kiện thuộc [-89.9; 88.8]. Nếu không khỏa bắt người dùng nhập lại đến khi nào khỏa điều kiện.

while có else

- Kết hợp **while** có **else** (mở rộng)

```
while <điều kiện kiểm tra> :
        <Lệnh hoặc khối lệnh>
else:
        <Lệnh hoặc khối lệnh trong else>
```

- Cách nhớ:

Khi kết thúc while thì mới thực hiện lệnh trong else
"nếu điều kiện trong while **không** đúng thì thực hiện lệnh trong else"

while có else

- ***Ví dụ minh họa***

```
x = float(input("Nhập x = "))
while x < 0:
    x = float(input("Nhập lại x = "))
else:
    print("Giá trị đã nhập:", x)
```

while có else

- ***Ví dụ minh họa***

```
count = 0
n = int(input("Nhập vào số lần cần lặp: "))
while (count < n):
    print ("Lần lập thứ:", count + 1,"\tBiến đếm:", count)
    count = count + 1
else:
    print("\nThực hiện lệnh trong else, do:",
          "\ncount = ", count, "\nn = ", n,
          "\nbool(count < n) = ", bool(count<n))
```

Kết quả:

```
Nhập vào số lần cần lặp: 5
Lần lập thứ: 1      Biến đếm: 0
Lần lập thứ: 2      Biến đếm: 1
Lần lập thứ: 3      Biến đếm: 2
Lần lập thứ: 4      Biến đếm: 3
Lần lập thứ: 5      Biến đếm: 4

Thực hiện lệnh trong else, do:
count =  5
n =  5
bool(count < n) =  False
```

break

- ***Giới thiệu***
  - Lệnh break được sử dụng trong vòng lặp để chấm dứt vòng lặp theo ý muốn.
  - **break** chủ yếu sử dụng trong vòng lặp **while** và **for**

break

- Dùng **break** để thoát **for**

```
for <biến vòng lặp> in <dãy giá trị> :
    <Lệnh hoặc khối lệnh bên trong vòng lặp>

    if <điều kiện để dừng vòng lặp> :
        break
    <Lệnh hoặc khối lệnh bên trong vòng lặp>
```

break

- ***Ví dụ***

```
tup = ("Đỏ","Cam","Vàng","Lục","Lam","Chàm","Tím")
for i in tup:
    print(i, "\t")
    if i == "Lục":
        break
```

Kết quả:

```
Đỏ
Cam
Vàng
Lục
```

break

- Dùng **break** để thoát **while**

```
while <điều kiện kiểm tra> :
        <Lệnh hoặc khối lệnh bên trong vòng lặp>
        if <điều kiện để dừng vòng lặp> :
                break
        <Lệnh hoặc khối lệnh bên trong vòng lặp>
```

continue

- ***Giới thiệu***
  - Lệnh **continue** được áp dụng trong vòng lặp **for** hoặc **while**.
  - Khi gặp lệnh **continue**, vòng lặp sẽ phớt lờ (bỏ qua) các lệnh phía sau continue chứ không dừng vòng lặp.
  - Vòng lặp chỉ dừng lại khi <điều kiện lặp> có giá trị False.

continue

- Sử dụng **continue** trong **for**

```
for <biến vòng lặp> in <dãy giá trị> :
        <Lệnh hoặc khối lệnh bên trong vòng lặp>
        if <điều kiện>          :
                continue
        <Lệnh hoặc khối lệnh bên trong vòng lặp>
```

continue

- Sử dụng **continue** trong **while**

```
while <điều kiện kiểm tra> :
        <Lệnh hoặc khối lệnh bên trong vòng lặp>
        if <điều kiện> :
                continue
        <Lệnh hoặc khối lệnh bên trong vòng lặp>
```

continue

- ***Minh họa***

```
tup = ('đỏ', 'cam', 'vàng', 'lục', 'lam', 'chàm', 'tím')

for i in tup:
    if i == 'vàng' or i == 'lam':
        continue
    print(i)
```

pass

- ***Giới thiệu***
  - Lệnh **pass** để đặt chỗ trước cho phần thân trong lớp, hàm, vòng lặp… đó là rỗng, không có lệnh để cho IDE không báo lỗi cú pháp
  - Cú pháp câu lệnh chỉ một từ khóa “pass”.

pass

- ***Ví dụ minh họa***

```
colors =["Đỏ", "Cam", "Vàng", "Lục", "Lam", "Chàm", "Tím"]
for i in colors:
    pass
```

Kết quả: Chương trình sẽ thông dịch, không thấy xuất hiện lỗi và phớt lờ khi thực hiện.

BÀI TẬP

BÀI #1

- Tạo một danh sách tự động các số chẵn nguyên và chia hết cho 5 từ 2018 đến 2828.

BÀI TẬP

BÀI #2

Viết chương trình tạo một danh sách tự động các số chẵn nguyên từ 2020 đến 3838.

1. Tạo danh sách các số chia hết cho 9 từ danh sách thu được.
2. Các số thu được sẽ được in thành chuỗi trên một dòng, cách nhau bằng 1 tab.

BÀI TẬP

BÀI #3

- Người dùng nhập một giá trị nguyên n từ bàn phím, hãy viết chương trình để tạo ra một dict chứa các phần tử theo cấu trúc sau:

$$i : i^i$$

  - i là số nguyên từ 1 đến n (bao gồm cả 1 và n)
  - In ra dict vừa tạo.

Ví dụ: Giả sử n là 3 thì đầu ra sẽ là: {1:1, 2:4, 3:27}.

PHẦN

BÀI TẬP TỔNG HỢP

(Các kiểu dữ liệu, cấu trúc rẽ nhánh, lặp)

BÀI TẬP TỔNG HỢP

BÀI #1

- Từ một chuỗi str_input được nhập vào từ bàn phím. Hãy viết lệnh thực hiện các chức năng sau:
  - Tính độ dài chuỗi.
  - Đếm và in các ký tự đặc biệt: ! @ # $ % ^ & * ( ) - = + . /
  - Đếm và in các ký tự chữ cái từ [a-z]
  - Đếm và in các ký tự chữ số từ [0-9]
  - Đếm và in các ký tự chữ [A-Z]

Font 0, pwritx_database

A A	B B	C C	D D	E E	F F	G G	H H	I I	J J
K K	L L	M M	N N	O O	P P	Q Q	R R	S S	T T
U U	V V	W W	X X	Y Y	Z Z	a a	b b	c c	d d
e e	f f	g g	h h	i i	j j	k k	l l	m m	n n
o o	p p	q q	r r	s s	t t	u u	v v	w w	x x
y y	z z	1 1	2 2	3 3	4 4	5 5	6 6	7 7	8 8
9 9	0 0	- –	= =	\ \	. ·	! !	@ @	# #	$ $
% %	^ ^	& &	* *	((	))	_ –	+ +	| |	~ ~
[[	]]	{ {	} }	; ;	' ‘	: :	" “	, ,	. .
/ /	< <	> >	? ?						

BÀI #2

Viết hàm kiểm tra để phát hiện chuỗi nhập vào có phải là một e-mail hay không?

- Chú ý: Nếu là các e-mail xuất phát từ Gmail, Yahoo, Hotmail, Outlook… (gọi tắt là tập luật tên miền e-mail) thì trước ký tự @ là một chuỗi tối thiểu 6 ký tự, không khoảng trắng và ký tự đặc biệt.
- Khảo sát và tìm thêm các tên miền e-mail để bổ sung vào tập luật giới hạn trên.

BÀI TẬP TỔNG HỢP

BÀI #3

Viết hàm kiểm tra quá trình đăng nhập ID user và password khi login vào hệ thống phải thõa mãn các tiêu chí sau:

- ID User:
  - Là một chuỗi, có ít nhất 6 ký tự, tối đa 24 ký tự.
  - Không chứa các ký tự: ! @ # $ % ^ & * ( ) - = +
  - Không khoảng trắng.
- Password:
  - Ít nhất 1 chữ cái nằm trong [a-z]
  - Ít nhất 1 số nằm trong [0-9]
  - Ít nhất 1 kí tự nằm trong [A-Z]
  - Ít nhất 1 ký tự nằm trong [$ # @]
  - Độ dài mật khẩu tối thiểu 6 ký tự
  - Độ dài mật khẩu tối đa 24 ký tự

BÀI TẬP TỔNG HỢP

BÀI #4

- Viết chương trình mô phỏng trò chơi **Kéo - Búa - Bao** giữa người và máy.

Qui ước:

- Kéo > Bao
- Búa > Kéo
- Bao > Búa

BÀI TẬP TỔNG HỢP

BÀI #5

- Nâng cấp từ **Bài 4** (nhiều người chơi tự động với nhau)
- Số lượng người được chọn ngẫu nhiên từ 8 đến 20 người.

BÀI TẬP TỔNG HỢP NÂNG CAO MINH HỌA VÉ SỐ VIETLOT 6/45

Người mua:

- Chọn 6 số bất kỳ từ 1 đến 45, các số không trùng nhau trong 1 vé. Gọi là 1 dãy số trên 1 vé. Ví dụ 10-18-26-33-39-44
- Có thể mua nhiều vé số tự chọn (n vé). Các vé có thể trùng dãy số với nhau.
- Giá bán mỗi vé 10.000 đ

Máy xổ số:

- Random 6 số bất kỳ từ 1 đến 45, các số không trùng nhau. Gọi là dãy số trúng thưởng.

Dò kết quả: So sánh dãy số người mua và dãy số trúng thưởng:

- Nếu trùng nhau 3 con số thì người chơi nhận: 30.000 đ
- Nếu trùng nhau 4 con số thì người chơi nhận: 300.000 đ
- Nếu trùng nhau 5 con số thì người chơi nhận: 10.000.000 đ
- Nếu trùng nhau 6 con số thì người chơi nhận: 10.000.000.000 đ

Thống kê kết quả người chơi nhận được bao nhiêu tiền khi dò kết quả.

Hôm nay, **kết thúc!**

- Nghi Hoàng Khoa
- khoanh@uit.edu.vn
- inseclab.uit.edu.vn