



BÀI 17: BIẾN VÀ LỆNH GÁN



Trong Đại số, người ta thường dùng chữ để thay thế cho số cụ thể, ví dụ hằng đẳng thức $(a + b)^2 = a^2 + 2ab + b^2$ đúng cho mọi giá trị a , b . Trong các ngôn ngữ lập trình, người ta cũng dùng các kí tự hoặc nhóm các kí tự (được gọi là biến (variable) hay biến nhớ) để thay cho việc phải chỉ ra các giá trị dữ liệu cụ thể.

Theo em , sử dụng biến có những lợi ích gì?





Quan sát các lệnh sau, n ở đây được hiểu là gì?

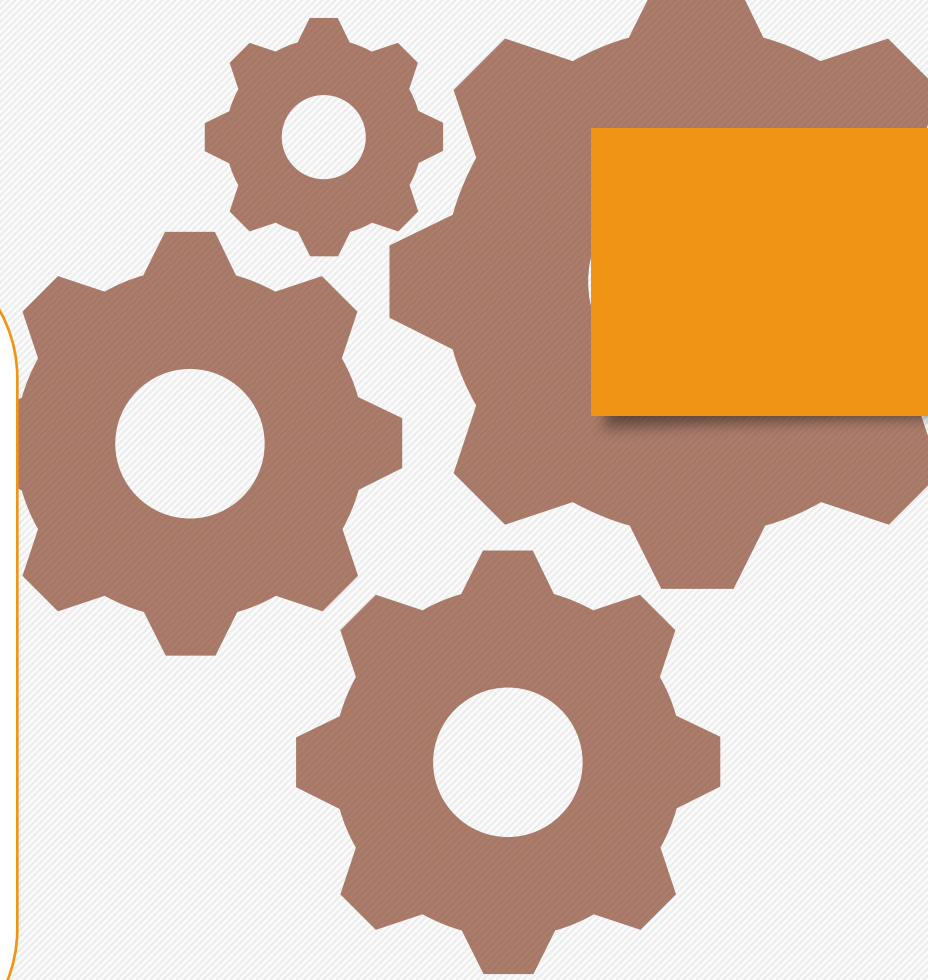
>>> $n = 5$

>>> $n \leftarrow$ Sau khi gán $n=5$ n sẽ được hiểu là đối tượng số nguyên có giá trị 5

5

>>> $n + 3$

8



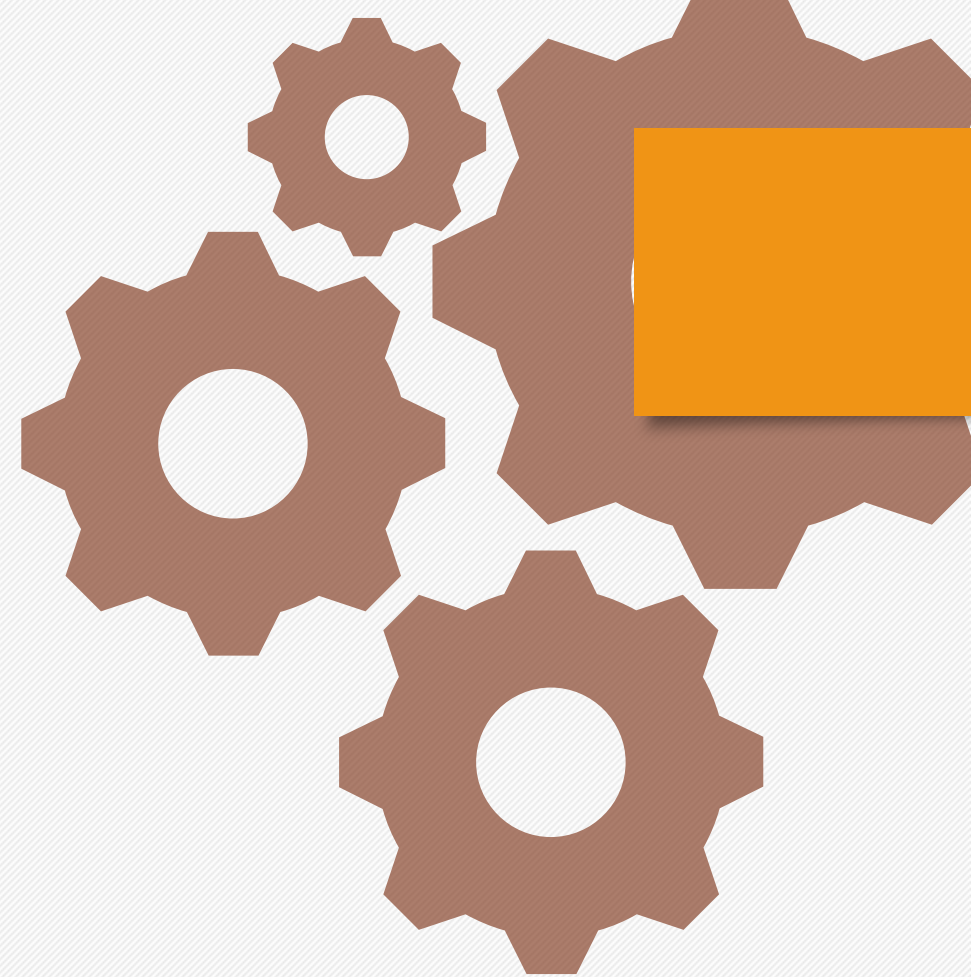


1. BIẾN VÀ LỆNH GÁN

- Biến là tên (định danh) của một vùng nhớ dùng để lưu trữ giá trị (dữ liệu) và giá trị đó có thể được thay đổi khi thực hiện chương trình.
- Biến trong Python được tạo ra khi thực hiện lệnh gán.
- Cú pháp của lệnh gán:

<biến> = <giá trị>

- Khi thực hiện lệnh gán, <giá trị> bên phải sẽ được gán cho <biến>. Nếu biến chưa được khai báo thì nó sẽ được khởi tạo khi thực hiện câu lệnh gán.
- Biến trong Python được xác định kiểu dữ liệu tại thời điểm gán giá trị nên không cần khai báo trước kiểu dữ liệu cho biến.



Ví dụ:

Shell x

```
>>> x = 5
```

```
>>> x    # x được gán 5 và có kiểu số nguyên tại đây
```

```
5
```

```
>>> y = "Tin học 10"
```

```
>>> y    # y được gán "Tin học 10" có kiểu xâu kí tự tại đây
```

```
'Tin học 10'
```

```
>>> |
```

Activate Windows

Go to Settings to activate Windows.

Python 3.7.9

- Có thể thực hiện tất cả các phép toán thông thường như: $+$, $-$, $*$, $/$, ... trên các biến có cùng kiểu dữ liệu.

Ví dụ:

```
>> x = y = 1  
>> x = y + 1  
>> z = (x+y)**x  
>> z
```

← Có thể gán đồng thời nhiều biến với một giá trị.

← Phép tính lũy thừa: $(x + y)^x$



- Có thể gán giá trị biểu thức cho biến. Cú pháp:

<biến> = <biểu thức>

- Khi thực hiện lệnh này, Python sẽ tính giá trị <biểu thức> và gán kết quả cho <biến> => mọi biến có trong <biểu thức> đều cần được xác định giá trị trước.

Ví dụ:



```
>>> x = 5
>>> y = 2
>>> z = x/y
>>> z
2.5
```

x là biến kiểu số nguyên có giá trị bằng 5.
y là biến kiểu số nguyên có giá trị bằng 2.
z là biến kiểu số thực có giá trị bằng 2.5.

KẾT NỐI TRI THỨC

- Tên biến thường được đặt sao cho dễ nhớ và có ý nghĩa.

Ví dụ:

```
>>> ten = "Hoài Nam"  
>>> print("Xin chào",ten)  
Xin chào Hoài Nam
```

- Có thể gán nhiều giá trị đồng thời cho nhiều biến. Cú pháp của lệnh gán đồng thời:

$\langle \text{var1} \rangle, \langle \text{var2} \rangle, \dots, \langle \text{varn} \rangle = \langle \text{gt1} \rangle, \langle \text{gt2} \rangle, \dots, \langle \text{gtn} \rangle$



Ghi nhớ:

- Biến là tên của một vùng nhớ dùng để lưu trữ giá trị (dữ liệu) và giá trị đó có thể được thay đổi khi thực hiện chương trình.
- Cú pháp lệnh gán:

<biến> = <biểu thức>

- Quy tắc đặt tên biến (định danh):
 - + Chỉ gồm các chữ cái tiếng Anh, các chữ số từ 0 đến 9 và kí tự gạch dưới “_”.
 - + Không bắt đầu bằng chữ số.
 - + Phân biệt chữ hoa và chữ thường.

1. Các tên biến nào dưới đây là hợp lệ trong Python?

- a. `_name` b. `12abc`
- c. `My country` d. `m123&b`
- e. `xyzABC`

2. Sau các lệnh dưới đây, các biến `x`, `y` nhận giá trị bao nhiêu?

```
>>> x = 10
```

```
>>> y = x**2 - 1
```

```
>>> x = x//2 + y%2
```

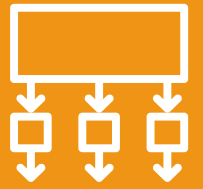
3. `a`, `b` nhận giá trị gì sau các lệnh sau ?

```
>>> a, b = 2, 3
```

```
>>> a, b = a+b, a - b
```



2. CÁC PHÉP TOÁN TRÊN MỘT SỐ KIỂU DỮ LIỆU CƠ BẢN



Ví dụ 1. Các phép toán trên dữ liệu kiểu số.

```
>>> a,b = 10,3
```

← Sau lệnh này thì $a = 10$, $b = 3$.

```
>>> (a+b)**2 + (a-b)*10
```

```
239
```

```
>>> (a//b)*b + a%b
```

```
10
```

```
>>> c = b/2
```

← Sau lệnh này, c là biến kiểu thực có giá trị 1.5.

```
>>> c
```

```
1.5
```



- Tất cả các phép toán đều được thực hiện từ trái sang phải, riêng phép lũy thừa (**) thì thực hiện từ phải sang trái.
- Các phép toán cơ bản với dữ liệu kiểu số (số thực và số nguyên) trong Python là phép cộng "+", trừ "-", nhân "*", chia "/", lấy thương nguyên "//", lấy số dư "%" và phép lũy thừa "**".
- Thứ tự thực hiện các phép tính như sau: phép lũy thừa ** có ưu tiên cao nhất, sau đó là các phép toán /, *, //, %, cuối cùng là các phép toán +, -.



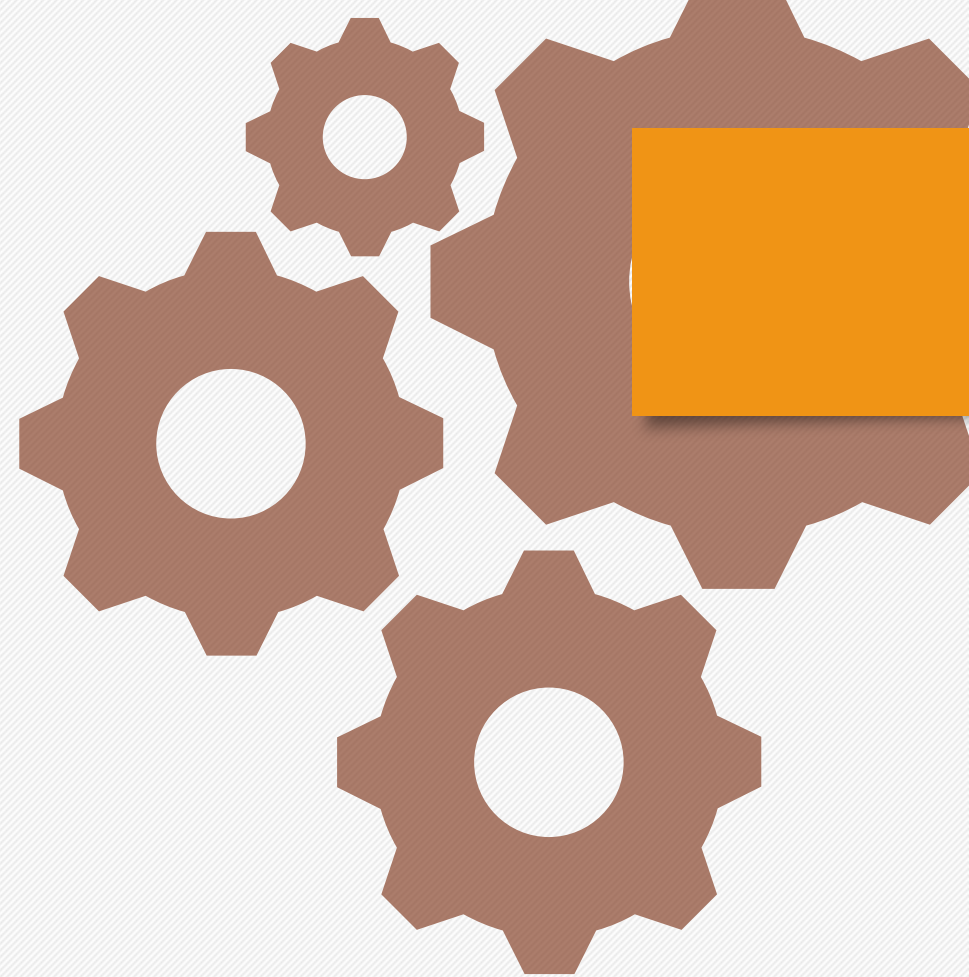
Ví dụ, lệnh sau :

```
>>> 3/2+4*2**4-5//2**2
```

tương đương với lệnh:

```
>>> 3/2+4 * (2**4) - 5//(2**2)
```

Chú ý. Nếu có ngoặc thì biểu thức trong ngoặc được ưu tiên thực hiện trước.



Ví dụ 2. Các phép toán với dữ liệu kiểu chuỗi ký tự

```
>>> s1 = "Hà Nội"
```

```
>>> s2 = "Việt Nam"
```

```
>>> s1 + s2
```

Phép nối + nối hai chuỗi ký tự.

```
" Hà Nội Việt Nam"
```

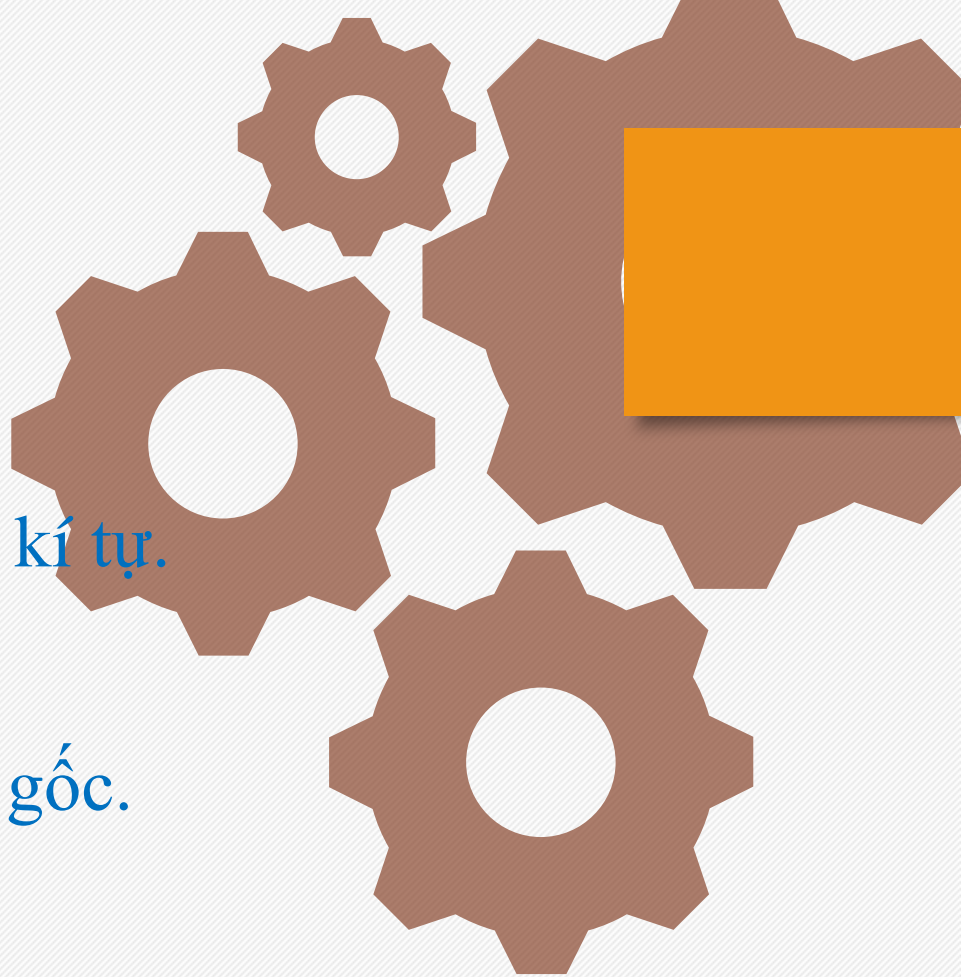
```
>>> "123" * 5
```

Phép * n lặp n lần chuỗi gốc.

```
"123123123123123"
```

```
>>> s*0 # Phép *n với số  $n \leq 0$  thì được kết quả là chuỗi rỗng.
```

Trong biểu thức có cả số thực và số nguyên thì kết quả sẽ có kiểu số thực



Ghi nhớ:

- Các phép toán trên dữ liệu kiểu số: +, -, *, /, //, %, **.
- Các phép toán trên dữ liệu kiểu xâu: + (nối xâu) và * (lặp)





1. Mỗi lệnh sau là đúng hay sai? Nếu đúng thì cho kết quả là bao nhiêu?

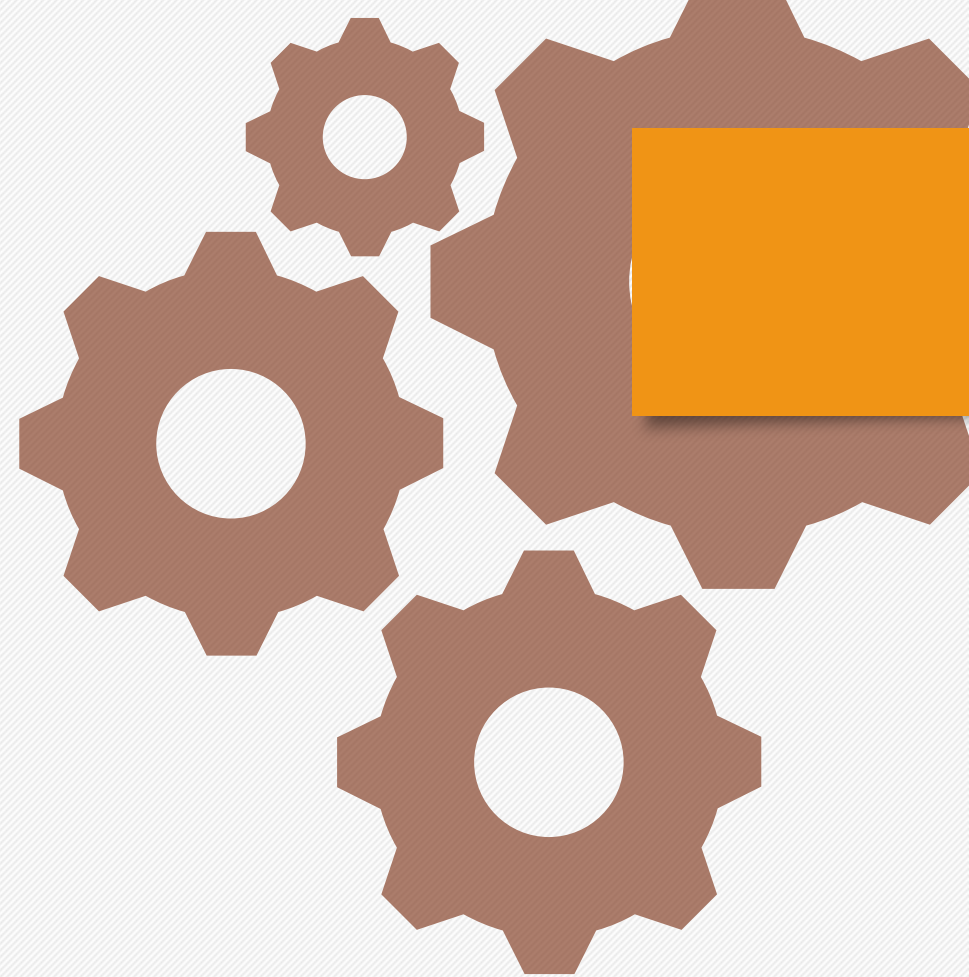
>>> (12- 10//2) **2- 1

>>> (13 + 45**2) (30//12 - 5/2)

2. Mỗi lệnh sau cho kết quả là xâu kí tự như thế nào?

>>> “”*20 + “010”

>>> “10” + “0” *5



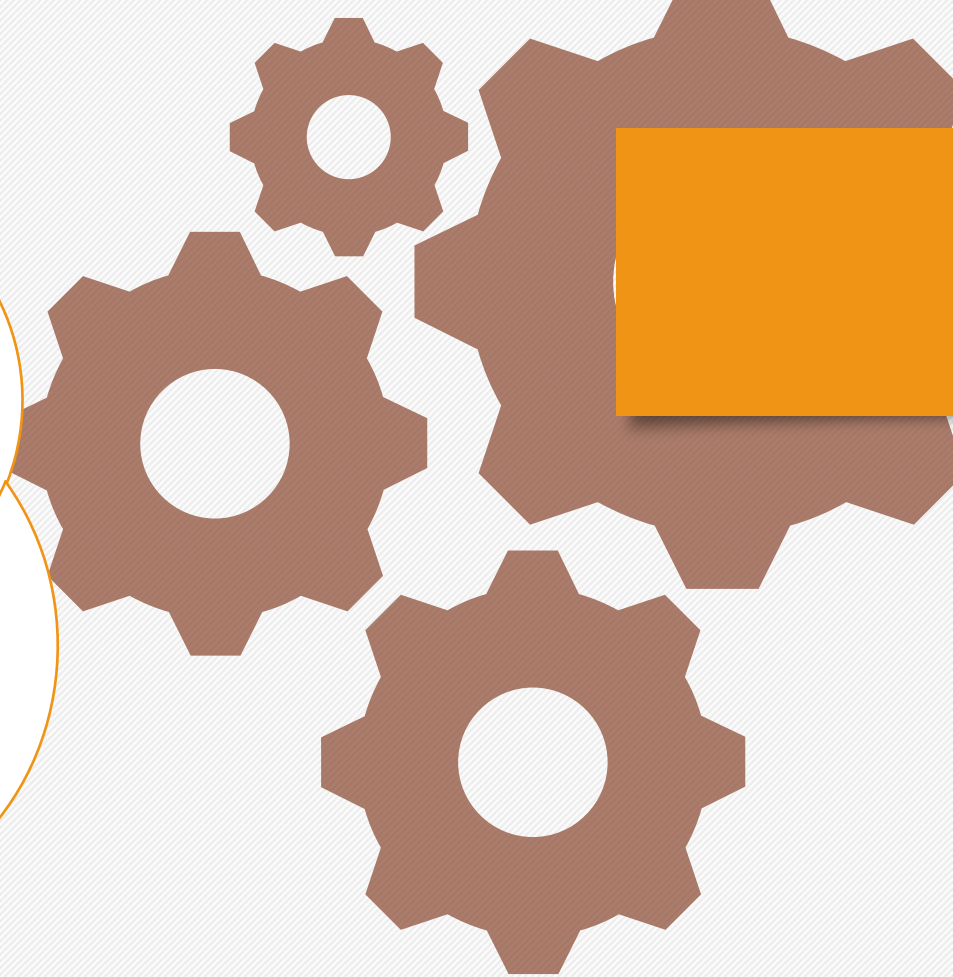
Quan sát các lệnh sau, tìm hiểu
vì sao Python báo lỗi

```
>>> if = 12
```

SyntaxError: invalid syntax

```
>>> with = "Độ rộng"
```

SyntaxError: invalid syntax

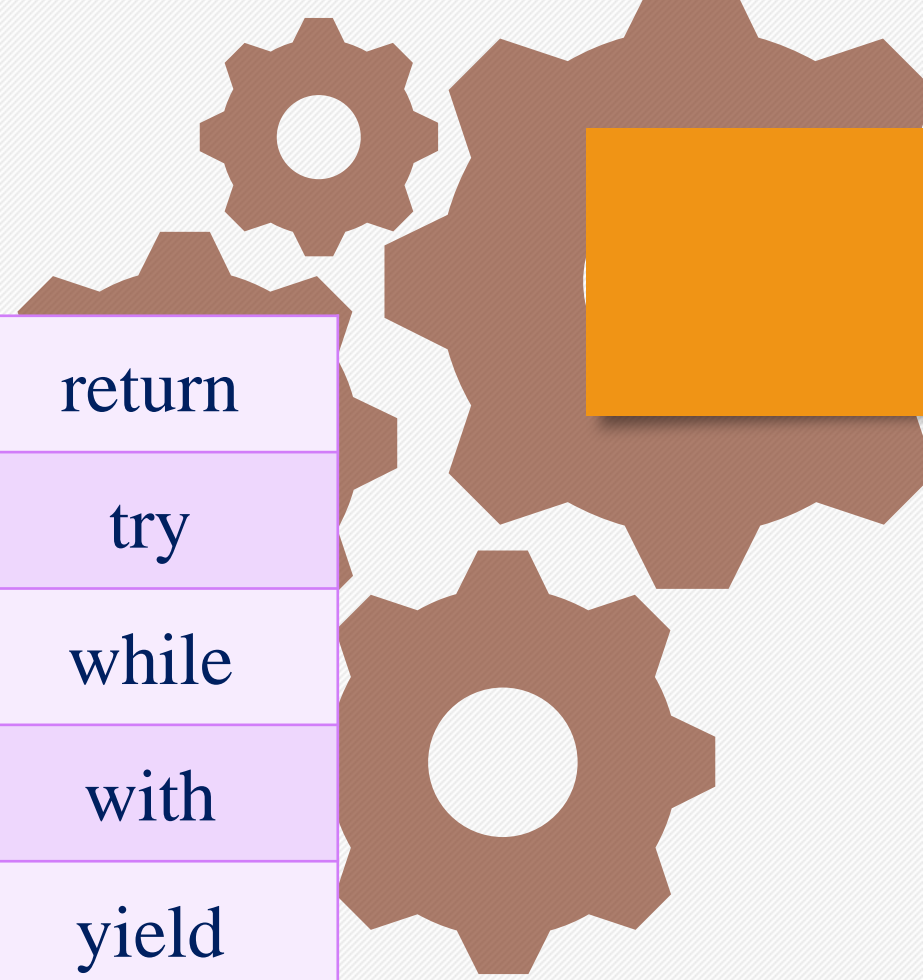


3. TỪ KHÓA

- Một tập hợp các từ tiếng Anh đặc biệt được sử dụng vào mục đích riêng của ngôn ngữ lập trình, được gọi là các **từ khóa (keyword)** của ngôn ngữ lập trình. Khi viết chương trình không được đặt tên biến hay các định danh trùng với từ khóa.

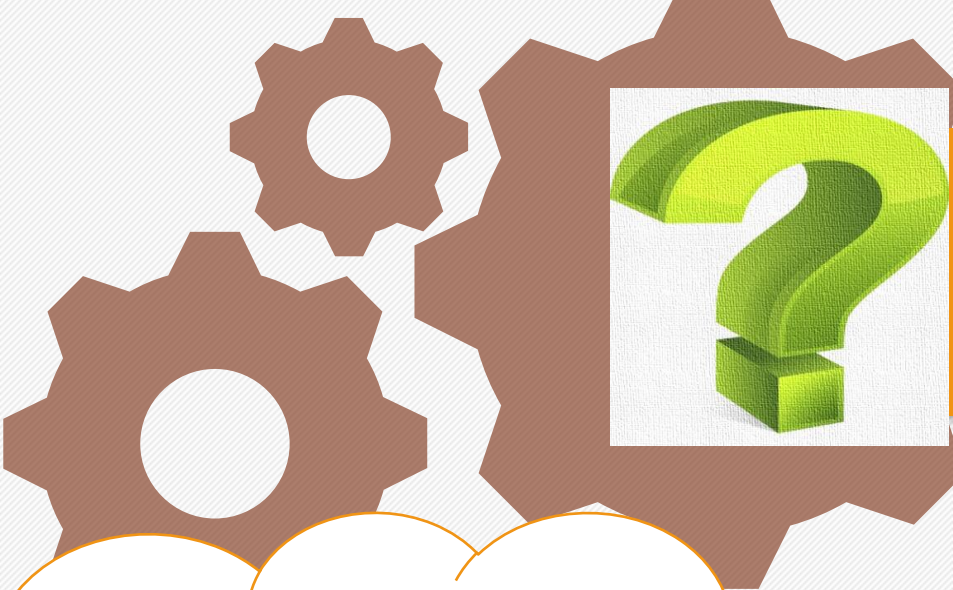
- Một số từ khóa trong Python phiên bản 3.x.

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	



Ghi nhớ

- *Từ khóa* là các từ đặc biệt tham gia vào cấu trúc của ngôn ngữ lập trình
- Không được phép đặt tên biến hay các định danh trùng với từ khóa



? Các tên biến sau có hợp lệ không?

a) _if

b) global

c) nolocal

d) return

e) true

4. THỰC HÀNH

Tạo và làm việc với biến, tính toán với các kiểu dữ liệu cơ bản trong Python.

Nhiệm vụ 1. Thực hiện các phép tính sau trong môi trường lập trình Python, so sánh kết quả với việc tính biểu thức toán học.

a) $(1+2+3+\dots+10)^3$

b) $1/2 + 1/3 + 1/4 + 1/5$

c) Thực hiện lệnh gán $x = 2$, $y = 5$ rồi tính giá trị biểu thức $(x + y)(x^2 + y^2 - 1)$

d) Thực hiện gán $a = 2$, $b = 3$, $c = 4$ rồi tính giá trị biểu thức $(a + b + c)(a + b - c)$

Hướng dẫn: Các phép tính trên có thể thực hiện trong môi trường lập trình Python như sau

```
>>> (1+2+3+4+5+6+7+8+9+10)**3
```

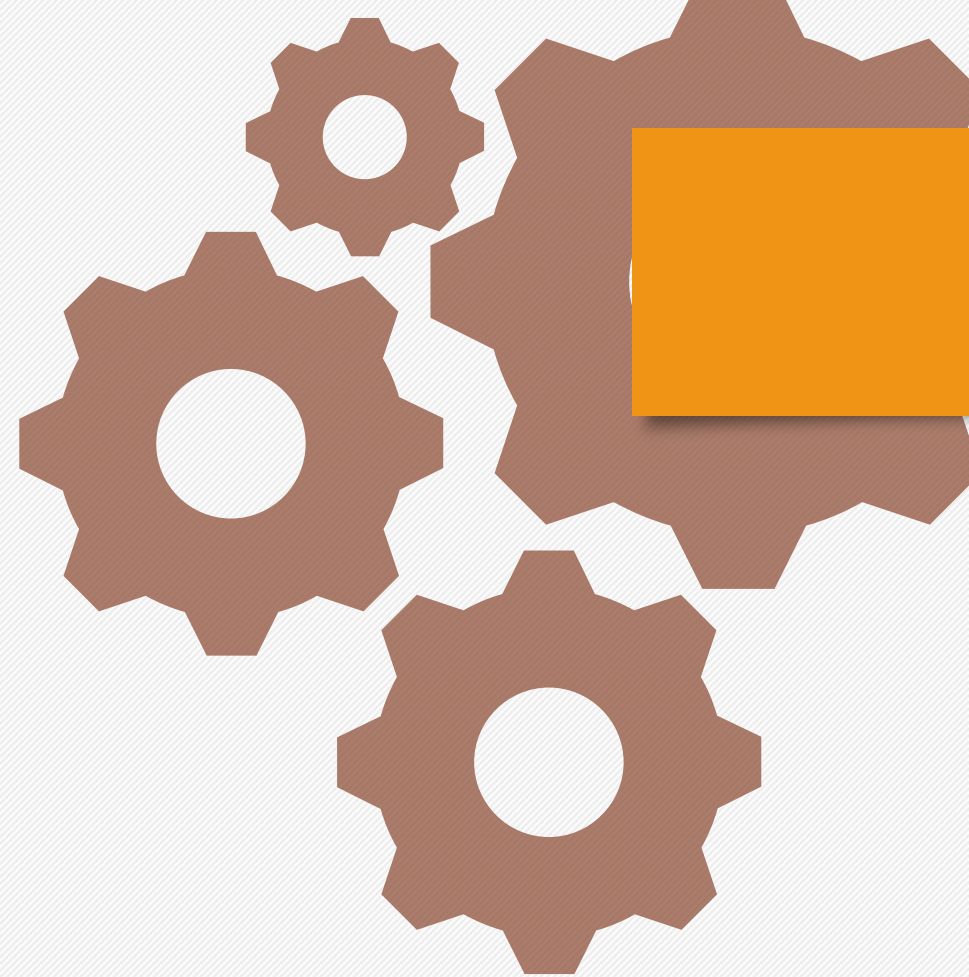
```
>>> x, y = 2, 5
```

```
>>> (x+y)*(x**2+y**2-1)
```

```
>>> 1/2 + 1/3 + 1/4 + 1/5
```

```
>>> a,b,c = 2,3,4
```

```
>>> (a+b+c) * (a+b-c)
```



Nhiệm vụ 2: Gán giá trị cho biến R là bán kính hình tròn rồi viết chương trình tính và in ra kết quả theo mẫu

Chu vi hình tròn là:

Diện tích hình tròn là:

Hướng dẫn: Soạn thảo chương trình sau trong môi trường lập trình Python

$R = 4.5$

$Pi = 3.14$

```
print("Chu vi hình tròn là:", 2*R*pi)
```

```
print("Diện tích hình tròn là:", pi*R*R)
```

Thực hiện chương trình và kiểm tra kết quả, so sánh với chế độ gõ lệnh trực tiếp



Em hãy điền đúng/sai cho các tên biến sau?

a) n, delta, x1, t12, Trường_sa Đ

b) 12t S

c) A b S

d) Ab Đ

e) AB Đ



Hãy cho biết giá trị lần lượt của 2 biểu thức sau trong Python: $(3 + 5) * 2 + 1$ và $3 + 5 * 2 + 1$

a) 17

b) 13

c) 24

d) 14



Hãy chuyển biểu thức toán học sang Python

Toán học	Python
$2a + 3b$	$2*a + 3*b$
$xy : z$	$x*y / z$
$b^2 - 4ac$	$b*b - 4*a*c$
$(a : b) c$	$(a/b)*c$

LUYỆN TẬP

1. Lệnh sau có lỗi gì?

```
>>> x = 1
```

```
>>> 123a = x + 1
```

SyntaxError: invalid syntax

2. Lệnh sau sẽ in ra kết quả gì?

```
>>> print("đồ rê mi " * 3 + "pha son la si đô " * 2)
```

3. Viết các lệnh để thực hiện việc đổi số giây ss cho trước sang số ngày, giờ, phút, giây, in kết quả ra màn hình.

Ví dụ, nếu $ss = 684\ 500$ thì kết quả in ra như sau:

684 500 giây = 7 ngày 22 giờ 8 phút 20 giây

Gợi ý. Sử dụng các phép toán lấy thương nguyên, lấy số dư và các cách đổi sau:

1 ngày = 86 400 giây; 1 giờ = 3 600 giây; 1 phút = 60 giây.

4. Hãy cho biết trước và sau khi thực hiện các lệnh sau, giá trị các biến x, y là bao nhiêu? Em có nhận xét gì về kết quả nhận được?

>>> x, y = 10, 7

>>> x, y = y, x



BÀI TẬP VỀ NHÀ

Bài 1: Em hãy nêu 3 tên biến đúng, 3 tên biến sai. Với tên biến sai, em hãy giải thích tại sao đó không phải là tên biến

Bài 2:

- 1) Ở cửa sổ Code, em hãy soạn thảo chương trình như trong hình bên, chạy chương trình và cho biết kết quả hiển thị trên màn hình
- 2) Thực hiện từng lệnh trong hình bên ở cửa sổ shell. Sau đó hãy thay phép nhân bằng một phép toán khác và xem kết quả

```
File Edit Format Run Options Window H
a = 123
b = 5
c = a * b
print(c)
```



BÀI TẬP

Bài 3: Em hãy hoàn thiện chương trình ở hình bên dưới bằng cách viết biểu thức gán cho biến pound để nhận được chương trình chuyển đổi đơn vị đo khối lượng từ đơn vị ki-lô-gam sang pound, biết rằng 1 kg bằng 2,205 pound. Em hãy thay đổi giá trị gán cho biến kilo để chạy thử nghiệm chương trình.

```
File Edit Format Run Options Window Help
kilo = 4.5
pound =
print (pound)
```



BÀI TẬP

Bài 4: Một mảnh vườn trồng cúc đại đóa có chiều rộng m mét, chiều dài n mét. Mỗi mét vuông trồng được một khóm hoa. Mỗi khóm hoa bán được a nghìn đồng. Em hãy viết chương trình để đưa ra màn hình tổng số tiền thu được khi bán hết hoa trong vườn. Hãy chạy chương trình với bộ dữ liệu đầu vào $m = 5, n = 18, a = 30$



BÀI TẬP

Bài 5: Xét đoạn chương trình ở hình bên. Em hãy cho biết c hay d nhận giá trị lớn hơn

```
File Edit Format Run Options Window Help
a = 15.8
b = 6.2
c = a // b
d = a % b
|
```

Bài 6: Có thể lưu chương trình Python dưới dạng tệp hay không?



BÀI TẬP

Bài 6: Em hãy hoàn thiện chương trình ở hình bên dưới bằng cách viết biểu thức gán cho biến pound để nhận được chương trình chuyển đổi đơn vị đo khối lượng từ đơn vị ki-lô-gam sang pound, biết rằng 1 kg bằng 2,205 pound. Em hãy thay đổi giá trị gán cho biến kilo để chạy thử nghiệm chương trình.

```
File Edit Format Run Options Window Help
kilo = 4.5
pound =
print (pound)
```