BÀI 17 BIẾN VÀ LỆNH GÁN

SAU BÀI NÀY EM SẼ:

- Biết cách thiết lập biến. Phân biệt được biến và từ khoá.
- Biết sử dụng lệnh gán và thực hiện một số phép toán trên kiểu số nguyên, số thực và xâu kí tự.

Trong Đại số, người ta 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) để thay cho việc phải chỉ ra các giá trị dữ liệu cụ thể.

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

1. BIẾN VÀ LỆNH GÁN

Hoạt động 1 Tìm hiểu khái niệm biến và lệnh gán

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

```
>>> n = 5
>>> n
5
>>> n + 3
8
```

Sau khi gán n = 5, n sẽ được hiểu là đối tượng số nguyên có giá trị 5.

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 như sau:

```
<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 như một số ngôn ngữ khác. Ví dụ:

```
>>> 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'
```

Sau câu lệnh gán, kiểu dữ liệu của biến được xác định. Cần ghi nhớ điều này để tránh thực hiện các phép toán giữa các biến có kiểu dữ liệu khác nhau. Chẳng hạn, nếu sau các lệnh trên, ta thực hiện lệnh z = x + y thì Python sẽ thông báo lỗi do không thể thực hiện phép cộng giữa dữ liệu kiểu số (biến x) với dữ liệu kiểu xâu kí tự (biến y).

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
9
```

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

Phép tính luỹ thừa: $(x + y)^x$

Ví dụ trên cho thấy có thể gán giá trị biểu thức cho biến. Câu lệnh gán có cú pháp tổng quát như sau:

```
<biến> = <biểu thức>
```

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

Ngoài việc gán giá trị trực tiếp cho biến, ta có thể gán giá trị cho biến thông qua tính toán giá trị của biểu thức với các biến đã được xác định 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.

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

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

Có thể gán nhiều giá trị đồng thời cho nhiều biến.

```
>>> x,y,z = 10,5,1
>>> x+y+z
16
```

Cú pháp của lệnh gán đồng thời như sau:

```
<var1>,<var2>,..., <varn> = <gt1>, <gt2>,..., <gtn>
```

Chú ý trong lệnh trên, số các biến bên trái bằng số các giá trị bên phải dấu "=".

- *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:
  - 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
```

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

Hoạt động 2 **Các phép toán trên dữ liệu kiểu số và kiểu xâu kí tự**

Tìm hiểu các phép toán trên dữ liệu kiểu số và kiểu xâu kí tự.

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

```
>>> a,b = 10,3
>>> (a+b)**2 + (a-b)*10
239
>>> (a//b)*b + a%b
10
>>> c = b/2
>>> c
1.5
```

Sau lệnh này thì a = 10, b = 3.

Phép toán // lấy thương nguyên của phép chia nguyên.
Phép toán % lấy số dư của phép chia nguyên.

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

Tất cả các phép toán đều được thực hiện từ trái sang phải, riêng phép luỹ thừa (**) thì thực hiện từ phải sang trái. Ví dụ, biểu thức 4**2**3 được thực hiện như sau 4**(2**3) = 4**8 = 65536.

```
>>> 4**2**3
65536
>>> (4**2)**3
4096
```

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 luỹ thừa "**".

Thứ tự thực hiện các phép tính như sau: phép luỹ 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 xâu kí tự.

```
>>> s1 = "Hà Nội"
>>> s2 = "Việt Nam"
>>> s1 + s2
'Hà NộiViệt Nam'
>>> "123"*5
'123123123123123'
>>> s*0
''
```

Phép + nối hai xâu kí tự.

Phép * n lặp n lần xâu gốc.

Phép * n với số n ≤ 0 thì được kết quả là xâu rỗng.

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

– 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
```

3. TỪ KHOÁ

Hoạt động 3 **Phân biệt biến và từ khoá**

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

```
>>> if = 12
SyntaxError: invalid syntax
>>> with = "Độ rộng"
SyntaxError: invalid syntax
```

Trong Python, cũng như mọi ngôn ngữ lập trình bậc cao, luôn có 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 khoá (keyword) của ngôn ngữ lập trình. Khi viết chương trình không được đặt tên biến hay các định danh trùng với từ khoá. Các lệnh trên, do đặt tên biến trùng với các từ khoá `if` và `with` nên bị báo lỗi.

Một số từ khoá trong Python phiên bản 3.x.

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

Từ khoá 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 khoá.

Các tên biến sau có hợp lệ không?

a) _if b) global c) nolocal d) return e) true

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) $(1 + 2 + 3 + ... + 10)^3$.

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.

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)
```

VẬN DỤNG

1. 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.

2. 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
```