



TIN HỌC 10



HỌ VÀ TÊN:

LỚP:

MÔN: TIN HỌC 10

NĂM HỌC 2022 -2023

BÀI 1. NGÔN NGỮ LẬP TRÌNH PYTHON

1. Lập trình

Khái niệm: Lập trình là sử dụng cấu trúc dữ liệu và các câu lệnh của **ngôn ngữ lập trình** cụ thể để mô tả dữ liệu và diễn đạt các thao tác của **thuật toán**.

Ví dụ:

Thuật toán	Chương trình
Bước 1: Nhập a, b. Bước 2: Hoán vị 2.1. $c \leftarrow a$ 2.2. $a \leftarrow b$ 2.3. $b \leftarrow c$ Bước 3: In giá trị a, b.	<pre>a=int(input('Nhập a: ')) b=int(input('Nhập b: ')) c=a a=b b=c print('Kết quả hoán vị: a=',a, 'và b=',b)</pre>

Câu hỏi: Học sinh hãy liệt kê từ ví dụ trên:

Câu 1. Thao tác của thuật toán và lệnh tương ứng của thao tác tại chương trình?

.....

.....

.....

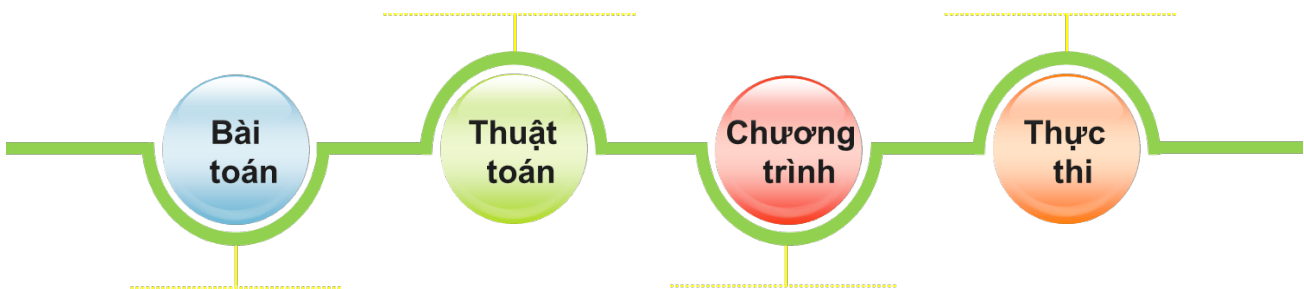
.....

Câu 2. Dữ liệu trong thuật toán và cấu trúc được mô tả trong chương trình?

.....

.....

.....



Hình 1 Các bước giải bài toán trên máy tính.

2. Ngôn ngữ lập trình.

- Khái niệm: Là ngôn ngữ để viết chương trình.

- Phân loại: chia làm 3 loại.

- Ngôn ngữ máy: thể hiện dưới dạng mã nhị phân để máy trực tiếp hiểu và thực hiện.
- Hợp ngữ: sử dụng kết hợp mã nhị phân và các từ viết tắt trong tiếng Anh.
- Ngôn ngữ bậc cao: sử dụng các ký hiệu gần với ngôn ngữ con người (ngôn ngữ tự nhiên).

3. Ngôn ngữ lập trình Python.

- Tác giả: Guido van Rossum (Hà Lan).
- Phát hành: 1991.
- Môi trường: Windows, Mac OS, Linux, Raspberry Pi, ...
- Là ngôn ngữ lập trình bậc cao, hỗ trợ: Artificial Intelligence – AI, Big Data – BG, Machine Learning – ML.

4. Chương trình dịch.

- **Khái niệm:** Là chương trình có chức năng chuyển đổi chương trình viết bằng ngôn ngữ bậc cao hoặc hợp ngữ thành chương trình thực hiện được trên máy tính.

- Thông dịch: (Interpreter)

- Kiểm tra chương trình nguồn.
- Chuyển đổi câu lệnh thành ngôn ngữ máy.
- Thực hiện lệnh vừa chuyển đổi.

- Biên dịch: (Compiler)

- Kiểm tra chương trình nguồn.
- Chuyển toàn bộ chương trình nguồn thành một chương trình đích có thể thực hiện được trên máy tính và có thể lưu trữ thành tập tin.

Câu hỏi:

Câu 1. Câu hỏi: Hãy cho biết ngôn ngữ lập trình được phân thành mấy loại? các ngôn ngữ C, Pascal, Python thuộc loại nào?

.....

.....

.....

.....

.....

Câu 2. Tại sao phải có chương trình dịch?

.....

.....

.....

Câu 3. Hãy cho biết các ngôn ngữ Pascal, C, Python sử dụng loại chương trình dịch nào?

5. Thành phần của ngôn ngữ lập trình

- Các thành phần của ngôn ngữ lập trình.

➤ Bảng chữ cái:

- Các ký tự trong bảng chữ cái : a-z, A-Z.
- 10 chữ số: 0 -> 9.
- Các ký tự đặc biệt: +, -, *, /, %, @, &...

➤ Cú pháp:

- ❖ Là bộ quy tắc viết chương trình.
- ❖ VD: Chỉ được dùng các ký tự theo hướng dẫn, có ký tự viết hoa, có ký tự viết thường, có lệnh phải lùi đầu dòng ...

➤ Ngữ nghĩa:

- ❖ Xác định ý nghĩa thao tác cần phải thực hiện, ứng với tổ hợp ký tự dựa vào ngữ cảnh của nó.

- Lưu ý:

- Python mặc định sử dụng bảng mã Unicode. Như vậy có thể sử dụng Tiếng Việt cho các đối tượng trong chương trình.

- Một số khái niệm:

- ❖ **Tên:** dùng để đại diện cho các **đối tượng** (*hằng, biến, ...*) trong chương trình.

❖ Quy tắc đặt tên trong Python:

- Dùng các ký tự thường (a-z), ký tự in hoa (A-Z), chữ số (0-9), dấu gạch dưới (_)
- Tên không bắt đầu bằng số, không dùng khoảng trắng (space) hoặc các ký hiệu đặc biệt như &, #, @... và có phân biệt chữ hoa, chữ thường.
- Không trùng với các từ khóa, tên hàm đã được ngôn ngữ lập trình sử dụng.
- Tên có độ dài không giới hạn, tuy nhiên cần đặt ngắn gọn và gọi nhớ tới đối tượng mà nó đại diện.

❖ Tên dành riêng.

- Tên dành riêng còn được gọi là từ khóa (keyword), được định nghĩa sẵn và không thể thay đổi chức năng.
- Tất cả các từ khóa đều được viết thường, trừ 3 từ khóa True, False, None.
- Ví dụ: else, await, import, pass, break, except, in, and, or...

❖ Tên chuẩn.

- Là những tên đã được định nghĩa thuộc thư viện chuẩn của Python (Module).
- Ví dụ: math

❖ Hằng và biến

- Hằng là đại lượng được đặt tên, dùng để lưu trữ giá trị và giá trị **không thể thay đổi** trong quá trình thực hiện chương trình.
- Biến là đại lượng được đặt tên, dùng để lưu trữ giá trị và giá trị **có thể được thay đổi** trong quá trình thực hiện chương trình.

❖ Chú thích.

- Là phần ghi chú trong chương trình dùng để ghi lại nội dung cần làm rõ về câu lệnh mà người lập trình cần, ghi chú không tham gia vào nội dung chương trình và không được biên dịch ra mã máy.
- Cách ghi chú thích: có 3 cách.
 - Dấu thăng (#) đầu dòng: dùng để ghi chú trên 1 dòng.
 - Ba dấu nháy đơn (') hoặc 3 nháy kép ("): dùng để ghi chú trên nhiều dòng.
- Ví dụ:

ghi chú trên một dòng.

''' ghi chú trên

nhiều dòng

'''

Câu hỏi:

Câu 1. Hãy liệt kê các loại ký tự được dùng trong ngôn ngữ Python?

.....

.....

.....

Câu 2. Ngữ nghĩa và cú pháp có vai trò gì trong việc viết chương trình bằng Python?

.....

.....

.....

.....

Câu 3. Hãy liệt kê các loại ký tự được phép dùng để đặt tên trong Python?

.....

.....

.....

Câu 4. Hãy cho biết tên chuẩn và tên dành riêng trong Python có đặc trưng gì?

.....

.....

.....

Câu 5. Hãy phân biệt sự khác nhau giữa hằng và biến?

.....

.....

.....

.....

.....

Câu 6. Chú thích được dùng để làm gì? Có mấy cách viết chú thích trong Python?

.....

.....

.....

.....

BÀI 2. KIỂU DỮ LIỆU, HẰNG, BIẾN VÀ CÂU LỆNH GÁN.

1. Kiểu dữ liệu.

Là việc phân loại theo tính chất các dữ liệu được dùng trong chương trình theo quy tắc của ngôn ngữ lập trình.

- Kiểu nguyên (int)

- Giá trị không giới hạn mà chỉ phụ thuộc vào bộ nhớ của máy tính.
- Khi gán một giá trị nguyên cho biến thì biến tự động có kiểu dữ liệu số nguyên.

```
# khai báo biến kiểu nguyên!

sohs = 40

soluongden = int(15.5)          # soluongden sẽ nhận giá trị
là 15
```

- Kiểu thực (float)

- Giá trị kiểu thực trong Python giới hạn 15 chữ số phần thập phân.
- Khi gán một giá trị số thực cho biến thì biến tự động có kiểu dữ liệu số thực.

```
# khai báo biến kiểu thực!

diemhkl = 7.5

tbmon = float(8)                # tbmon sẽ nhận giá trị là 8.0
```

- Kiểu chuỗi (string)

- Chuỗi là một dãy ký tự được đặt nằm trong cặp nháy kép (" ") hoặc nháy đơn ('...'), chuỗi rỗng là chuỗi không có ký tự nào.
- Khi gán một chuỗi cho biến thì biến tự động có kiểu dữ liệu chuỗi.

```
# khai báo biến chuỗi!

hovaten = 'Mai Sieu Phong'

bidanh = "Độc cô cầu bại!"

chiso = str(5)                  # gán chuỗi '5' cho biến chiso
```

- Các ký tự của chuỗi thuộc bảng mã Unicode nên có hỗ trợ tiếng Việt. Mỗi ký tự trong chuỗi được đánh thứ tự chỉ số 0,1,2, ... (theo hướng từ trái sang phải) và bắt đầu -1, -2, ... (theo hướng từ phải sang trái).
- Số ký tự trong chuỗi được gọi là độ dài của chuỗi đó.

- Kiểu logic (Bool)

- Trong Python kiểu logic có 2 giá trị **True** và **False**

2. Lệnh gán.

Trong Python lệnh gán được ký hiệu bởi dấu bằng (=)

Cú pháp: <tên đối tượng nhận> = <giá trị>

Đối tượng nhận có thể là hằng hoặc biến.

Ví dụ:

```
ten = 'An'
```

```
SISO = 45
```

3. Hằng

- Hằng là đại lượng được đặt tên, dùng để lưu trữ giá trị và giá trị không thể thay đổi trong quá trình thực hiện chương trình.

- **Lưu ý:** tên hằng được viết hoàn toàn bằng CHỮ HOA và dấu gạch dưới khi cần.

4. Biến.

- Biến là đại lượng được đặt tên, dùng để lưu trữ giá trị và giá trị có thể được thay đổi trong quá trình thực hiện chương trình.

- Cú pháp khai báo biến.

<Tên biến> = <Giá trị>

- Lưu ý:

- o Biến trong Python không cần phải khai báo trước, không nhất thiết phải khai báo kiểu dữ liệu mà khi gán dữ liệu cho biến Python tự động nhận dạng và tùy biến theo kiểu dữ liệu được gán.

- o Tên hằng, biến được đặt theo quy tắc.

❖ Quy tắc đặt tên trong Python:

- Dùng các ký tự thường (a-z), ký tự in hoa (A-Z), chữ số (0-9), dấu gạch dưới (_)
- Tên không bắt đầu bằng số, không dùng khoảng trắng (space) hoặc các ký hiệu đặc biệt như &, #, @... và có phân biệt chữ hoa, chữ thường.
- Không trùng với các từ khóa, tên hàm đã được ngôn ngữ lập trình sử dụng.
- Tên có độ dài không giới hạn, tuy nhiên cần đặt ngắn gọn và gợi nhớ tới đối tượng mà nó đại diện.

❖ Ví dụ: đặt tên biến

Tên đúng: a, b, x, y, chieudai, bankinh, ...

Tên sai: 1x, 2b, chieu rong, if, for, math.

- Cú pháp lệnh gán giá trị cho biến.

- Ví dụ: khai báo biến trong chương trình

# khai báo biến!	
<code>a = 'số năm'</code>	# biến a có kiểu xâu
<code>a = 5</code>	# biến a có kiểu số nguyên
<code>a = 5.8</code>	# biến a có kiểu số thực
<code>a = True</code>	# biến a có kiểu logic

- Ép kiểu giá trị khi gán cho biến.

# Ép kiểu giá trị khi gán!	
<code>b = str(10)</code>	# biến b có kiểu xâu
<code>b = int(5.8)</code>	# biến b có kiểu số nguyên
<code>b = float(5)</code>	# biến b có kiểu số thực

- Cú pháp câu lệnh gán nhiều giá trị cho nhiều biến.

<Tên biến 1>, <Tên biến 2>, ..., <Tên biến n> = <Giá trị 1>, <Giá trị 2>, ..., <Giá trị n>

Ví dụ: a, b, c = 100, "học lập trình", True

Như vậy:

a = 100

b = "học lập trình"

c = True

- Một số cách rút gọn lệnh gán.

Lệnh gán	Rút gọn
<code>x = 2</code> <code>y = 2</code> <code>z = 2</code>	<code>x = y = z = 2</code>
<code>a = 1</code> <code>b = 2</code> <code>c = 3</code>	<code>a, b, c = 1, 2, 3</code>
<code>x = a</code> <code>a = b</code> <code>b = x</code>	<code>a, b = b, a</code>

Câu hỏi:

Câu 1. Hãy nêu đặc điểm chính về cấu trúc của một chương trình viết bằng Python?

.....

.....

.....

.....

Câu 2. Hãy nêu đặc điểm chính của các loại dữ liệu cơ bản trong Python?

.....

.....

.....

.....

Câu 3. Trình bày cú pháp và chức năng của câu lệnh gán. Lấy ví dụ về khai báo biến theo các kiểu dữ liệu được trình bày ở trên?

.....

.....

.....

.....

.....

.....

.....

.....

Câu 4. Hãy lấy ít nhất 3 ví dụ khai báo biến cho mỗi kiểu dữ liệu được nêu trong bài học?

BÀI 3. PHÉP TOÁN VÀ BIỂU THỨC

1. Phép toán.

- Số học.

❖ Các phép toán với số nguyên:

Phép toán	Ký hiệu	Ví dụ
Cộng	+	$7 + 2 = 9$
Trừ	-	$7 - 2 = 5$
Nhân	*	$7 * 2 = 14$
Chia	/	$7 / 2 = 3.5$
Chia nguyên	//	$7 // 2 = 3$
Chia dư	%	$7 \% 2 = 1$
Lũy thừa	**	$7 ** 2 = 49$

❖ Các phép toán với số thực:

Phép toán	Ký hiệu	Ví dụ
Cộng	+	$7.0 + 2.0 = 9.0$
Trừ	-	$7.0 - 2.0 = 5.0$
Nhân	*	$7.0 * 2.0 = 14.0$
Chia	/	$7.0 / 2.0 = 3.5$
Lũy thừa	**	$25 ** 0.5 = 5.0$

- Quan hệ.

Phép toán	Ký hiệu	Ví dụ
Lớn hơn	>	$a > b$
Nhỏ hơn	<	$a < b$
Lớn hơn hoặc bằng	>=	$a >= b$
Nhỏ hơn hoặc bằng	<=	$a <= b$
Bằng	==	$a == b$
Khác	!=	$a != b$

- Logic.

Phép toán	Ý nghĩa
and	True khi cả 2 True
or	False khi cả 2 False
not	Phủ định

Bảng ví dụ kết quả phép toán logic

A	B	A and B	A or B	not A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

- Ghép chuỗi.

Cú pháp: **<xâu 1> + <xâu 2> + ... + <xâu n>**

Ý nghĩa: trả về 1 chuỗi là kết quả ghép của tất cả các chuỗi tham gia.

Ví dụ: *"học lập trình " + "Python " + "thật thú vị!"* → *"học lập trình Python thật thú vị!"*

2. Biểu thức.

- Biểu thức số học.

- ❖ Cấu tạo: Chữ số, phép toán số học và ngoặc tròn.
- ❖ Một số quy tắc thực hiện.
 - Các phép toán thực hiện theo độ ưu tiên theo bậc giảm dần từ trên xuống:
 1. Nhân, chia.
 2. Cộng, trừ.
 3. So sánh.
 4. Logic
 - Trong ngoặc tròn thực hiện trước.
 - Các phép toán cùng bậc thực hiện từ trái sang phải.
 - Khi viết biểu thức không bỏ qua dấu nhân.
- ❖ Ví dụ:

Toán học	Python
$5a + 6b$	$5*a + 6*b$
$\frac{xy}{z}$	$x*y/z$
$ax^2 + bx + c$	$a*x**2 + b*x + c$
$\frac{x+y}{x-\frac{1}{2}} - \frac{x-z}{xy}$	$(x+y)/(x-1/2) - (x-z)/(x*y)$

- Một số hàm số học chuẩn.

Tên hàm	Chức năng	Ví dụ
pi	tính giá trị số pi (3,14159 26535 89793)	$\sin(\pi) \rightarrow 1.2246467991473532e-16$
trunc(x)	tính phần nguyên của x	$\text{trunc}(3.5) \rightarrow 3$
round(x,n)	làm tròn x đến n chữ số thập phân	$\text{round}(50.25556, 2) \rightarrow 50.26$
abs(x)	tính giá trị tuyệt đối của x	$\text{abs}(-40) \rightarrow 40$
sqrt(x)	tính căn bậc 2 của x	$\text{sqrt}(100) \rightarrow 10.0$
max(a ₁ ,a ₂ ,a ₃ , ..., a _n)	tìm giá trị lớn nhất trong a ₁ ,a ₂ ,a ₃ , ..., a _n	$\text{max}(0, 100, -200) \rightarrow 100$
min(a ₁ ,a ₂ ,a ₃ , ..., a _n)	tìm giá trị nhỏ nhất trong a ₁ ,a ₂ ,a ₃ , ..., a _n	$\text{min}(0, 100, -200) \rightarrow -200$
sin(x)	tính sin của góc x (radian)	$\sin(\pi/2) \rightarrow 1.0$
cos(x)	tính cos của góc x (radian)	$\cos(2*\pi) \rightarrow 1.0$
tan(x)	tính tan của góc x (radian)	$\tan(\pi/2) \rightarrow 1.633123935319537e+16$
log(x)	tính log của x	$\log(10.15) \rightarrow 2.3174737054877963$
log(a, b)	Tính logarit của a cơ số b	$\text{Log}(8, 2) \rightarrow 3.0$
radians(x)	đổi x từ độ thành radians	$\text{radians}(60) \rightarrow 1.0471975511965976$
pow(x, y)	trả về giá trị của x ^y	$\text{pow}(2, 4) \rightarrow 16.0$
factorial(x)	Tính giai thừa của số nguyên x	$\text{Factorial}(5) \rightarrow 120$
gcd(a,b)	Tìm ước chung lớn nhất của 2 số a và b	$\text{gcd}(75, 50) \rightarrow 25$

Lưu ý: để sử dụng các hàm toán học trong chương trình thì phải khai báo thư viện *math* theo một trong các cú pháp:

```
import math      math.pi

from math import pi, trunc

from math import *
```

- Một số cách viết biểu thức thu gọn.

Biểu thức	Rút gọn
$n = n + a$	$n += a$
$n = n - a$	$n -= a$
$n = n * a$	$n *= a$
$n = n / a$	$n /= a$
$n = n // a$	$n //= a$
$n = n ** a$	$n ** = a$
$n = n \% a$	$n \% = a$

- Biểu thức quan hệ.

- Cú pháp: **<giá trị 1> <phép toán quan hệ> <giá trị 2>**

- Ví dụ:

$$a > b$$

$$x+5 <= y$$

$$n != x/y$$

- Biểu thức logic.

- Cú pháp: **<giá trị 1> <phép logic> <giá trị 2>**

Ví dụ: Với $x = 2, y = 5$

$$(x < 6) \text{ and } (y != 3) \rightarrow \text{True}$$

$$\text{not } (y > 3) \rightarrow \text{False}$$

- Bài tập minh họa sử dụng phép toán và biểu thức:

Cho số tự nhiên a có 3 chữ số, hãy viết biểu thức tính tổng 3 chữ số của a ?

(VD: $a = 428$ thì tổng 3 chữ số của a là: $4 + 2 + 8 = 14$)

Hướng dẫn thực hiện:

```
donvi = a % 10
x = a // 10
chuc = x % 10
tram = x // 10
tong = donvi + chuc + tram
```

Câu hỏi:

Câu 1. Cho 5 số nguyên a, b, c, d, e . Hãy viết biểu thức tính tổng giá trị của số hạng nhỏ nhất và số hạng lớn nhất?

.....

.....

.....

.....

Câu 2. Cho đường tròn có bán kính R . Hãy viết biểu thức tính chu vi C và diện tích S của đường tròn đó (làm tròn đến 2 chữ số phần thập phân).

.....

.....

.....

.....

Câu 3. Cho 3 số nguyên dương a, b, c . Hãy viết biểu thức tìm ước chung lớn nhất của 3 số đó.

BÀI 4. VÀO/ RA DỮ LIỆU TỪ THIẾT BỊ

1. Đưa dữ liệu ra màn hình.

- **Cú pháp:** `print(<danh sách kết quả>)`

➤ Giải thích:

- <danh sách kết quả> được viết dưới dạng <kết quả 1>, <kết quả 2>, ..., <kết quả n>. Các kết quả được viết cách nhau bởi dấu phẩy.
- Kết quả có thể là hằng, biến hoặc biểu thức.
- Các kết quả mặc định sẽ in trên cùng 1 dòng và cách nhau 1 khoảng trắng.
- Khi in xong các kết quả hàm sẽ mặc định xuống dòng và về đầu dòng mới.
- Nếu gắn thêm ký tự “\n” sau kết quả thì sau khi in kết quả đó xong lệnh sẽ xuống dòng.

➤ Ví dụ 1: in một kết quả là hằng ra màn hình.

```
print('Chào mừng đến với Python 3.10')
```

Kết quả:

Chào mừng đến với Python 3.10

➤ Ví dụ 2: in giá trị một biến sử dụng ký tự xuống dòng “\n”.

```
a = 5
print('In giá trị:\n')
print('a=', a)
```

Kết quả:

In giá trị:

a = 5

➤ Ví dụ 3: in giá trị một biểu thức

```
a = 5
b = 7
print('a + b =', a+b)
```

Kết quả:

a+b=12

Lưu ý: khi đưa nhiều kết quả ra màn hình ta sử dụng dấu phẩy để phân cách các kết quả, khi in ra các kết quả mặc định cách nhau 1 khoảng trắng.

- **Lệnh `print()` với các tham số.**

❖ **Tham số sep:** khi sử dụng lệnh `print()` để đưa nhiều kết quả ra màn hình thì tham số **sep** dùng để quy định ký tự phân cách giữa các kết quả. Trong trường hợp không viết thì **sep** mặc định là khoảng trắng, **sep** = `"\n"` thì sau khi in kết quả sẽ tự động xuống dòng, **sep** = `" "` tức rỗng thì giữa các kết quả không có khoảng cách.

➤ **Cú pháp:** `print(<kết quả 1>, ..., <kết quả n>, sep = [nội dung giữa các kết quả])`

➤ Ví dụ 1: sep mặc định là khoảng trắng

```
a, b, c = 100, 'nvc', True # gán nhiều giá trị cho nhiều biến
print(a,b,c)
```

Kết quả:

100 nvc True

➤ Ví dụ 2: sep là ký tự xuống dòng.

```
a, b, c = 100, 'nvc', True # gán nhiều giá trị cho nhiều biến
print(a,b,c, sep= "\n")
```

Kết quả:

100

Nvc

True

➤ Ví dụ 3: sep là ký tự - -

```
a, b, c = 100, 'nvc', True # gán nhiều giá trị cho nhiều biến
print(a,b,c, sep = "--")
```

Kết quả:

100--nvc--True

❖ **Tham số end:** khi sử dụng lệnh `print` để in kết quả ra màn hình thì sau khi in xong lệnh sẽ mặc định xuống dòng và về đầu dòng mới. Điều này xảy ra là do tham số **end** mặc định là ký tự xuống dòng (`"\n"`). Để thay đổi điều này ta điều khiển bằng tham số **end**.

➤ **Cú pháp:**

`print(<danh sách kết quả>, end = [nội dung cuối kết quả])`

Ví dụ 1: lệnh `print` với tham số **end** mặc định là xuống dòng

```
print("a", "b")
```

```
print("c")
```

Kết quả:

a b

c

Ví dụ 2: lệnh print với tham số end được điều khiển.

```
print("a", end = "=")  
print("b", end = "+")  
print("c", end = "\n")  
print("học lập trình python")
```

Kết quả:

a=b+c

học lập trình python

2. Nhập dữ liệu từ bàn phím.

- Cú pháp: input()

Hàm dùng để trả về một xâu ký tự nhập từ bàn phím.

Ví dụ: nhập một xâu ký tự từ bàn phím và gán cho biến a

```
a = input( )  
print(a)
```

Kết quả:

Hoc lap trinh python

Hoc lap trinh python

Như vậy khi lệnh **input** được thực thi người dùng sẽ nhập từ bàn phím một xâu, xâu này sẽ được gán cho biến a và khi in biến a ta sẽ in nội dung vừa nhập.

- Lệnh input có thêm phần hướng dẫn:

Phần này tương tự hàm print có thể in ra màn hình một xâu dùng để hướng dẫn người dùng nhập thông tin cho đúng mục đích.

Cú pháp: **input([<lời hướng dẫn>])**

Ví dụ:

```
a = input("nhập một số nguyên cho a:" )  
print(a)
```

Kết quả:

nhập một số nguyên cho a: 5

5

- Hàm input sử dụng ép kiểu dữ liệu:

Do hàm input() trả về một chuỗi ký tự nên khi muốn nhận kết quả trả về là một kiểu dữ liệu khác (số nguyên, số thực ...) ta phải sử dụng phương pháp ép kiểu.

Cú pháp: **<kiểu dữ liệu ép>(input())**

Ví dụ: ép kiểu số nguyên

```
a = int(input("Nhập a:"))
b = int(input("Nhập b:"))
print(a,b, sep = "+", end = "=")
print(a+b)
```

Kết quả:

Nhập a: 5

Nhập b: 4

5+4=9

Lưu ý: int ép kiểu về số nguyên, float ép kiểu về số thực, bool ép kiểu về kiểu logic.

Câu hỏi:

Câu 1. Hãy dùng lệnh `print( )` để in ra màn hình các hình vẽ sau:

a. Hình chữ nhật:

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

b. Hình cây thông:

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

Câu 2. Sử dụng hàm `print` và hàm `input` nhập từ bàn phím chiều dài 2 cạnh của một hình chữ nhật, sau đó tính và in diện tích, chu vi của hình chữ nhật đó ra màn hình.

Câu 3. Hãy giải thích ý nghĩa các lệnh và cho biết kết quả của chương trình sau:

```
x = 100
y = int(input())
```

```
z = 30
print(x, y, z, sep = "*", end = "=")
print(x*y*z)
```

.....

.....

.....

.....

.....

.....

.....

BÀI 5. BÀI TẬP THỰC HÀNH 1

1. Bài tập 1. Bài tập mẫu hướng dẫn thực hành

Viết chương trình nhập vào bán kính của một hình tròn từ bàn phím. Tính diện tích, chu vi (làm tròn đến 2 chữ số thập phân) của hình tròn đó và in ra màn hình.

- Các bước thực hiện chương trình:

a) Xây dựng thuật toán.

Bước 1: Nhập bán kính (r) của đường tròn từ bàn phím.

Bước 2: Tính chu vi đường tròn theo công thức $cv = 2\pi r$.

Bước 3: Tính diện tích đường tròn theo công thức $dt = \pi r^2$

Bước 4: In chu vi và diện tích đường tròn ra màn hình.

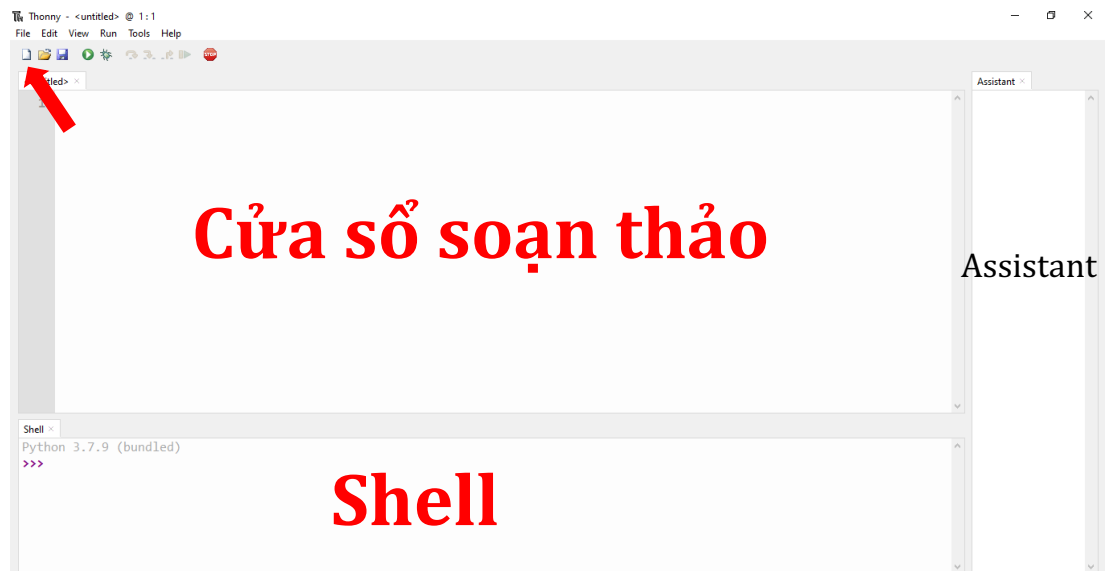
b) Khởi động trình biên tập.

✓ Tại Desktop phải chuột vào biểu tượng phần mềm **Thony**  chọn lệnh **Open**

✓ Tại giao diện phần mềm Thony chọn lệnh **File/ New** để tạo mới một tập tin để soạn thảo lệnh. (nếu chương chưa có file mới)

Cửa sổ chương trình soạn thảo xuất hiện có 3 cửa sổ con:

1. Cửa sổ soạn thảo.
2. Cửa sổ hiện thị kết quả. (Shell)
3. Cửa sổ hướng dẫn. (Assistant)



c) Soạn thảo chương trình

Học sinh thực hiện soạn thảo chương trình dưới đây vào cửa sổ số 1.

```
from math import pi

r = float(input("Nhập bán kính đường tròn r="))

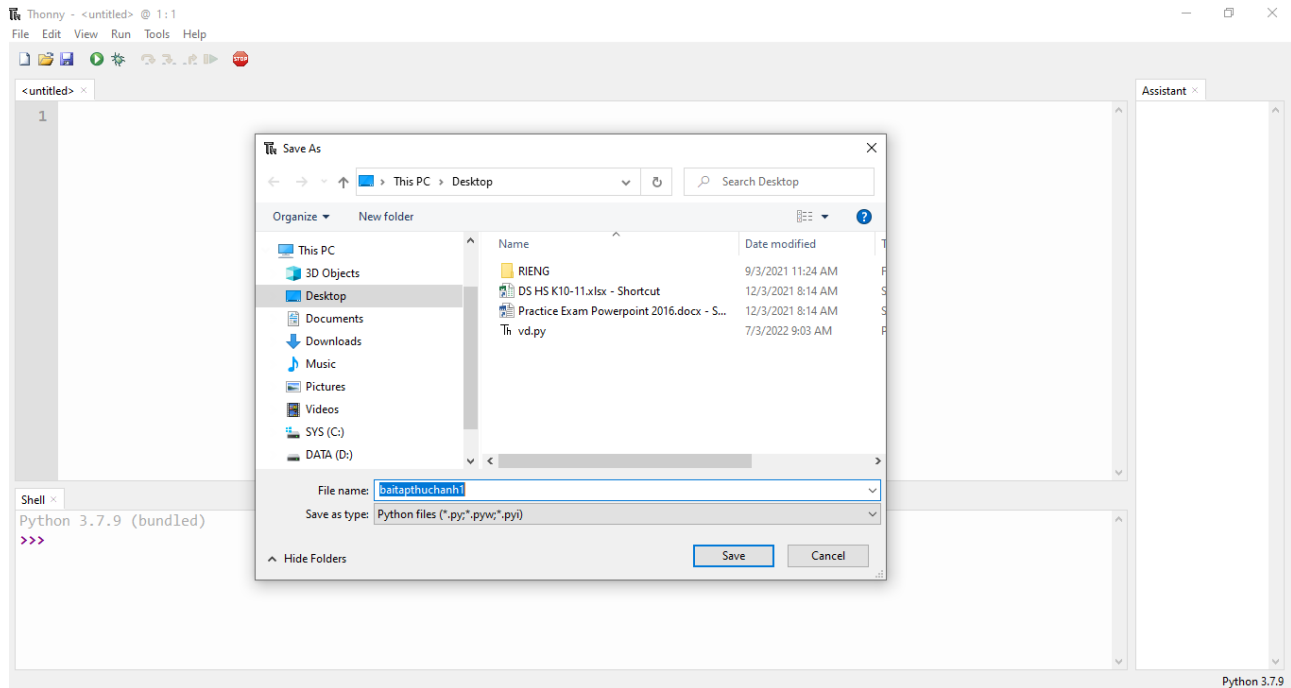
cv = round(2 * pi * r, 2)

dt = round(pi * r**2, 2)
```

```
print("Chu vi đường tròn là:", cv, "\nDiện tích đường tròn là:", dt)
```

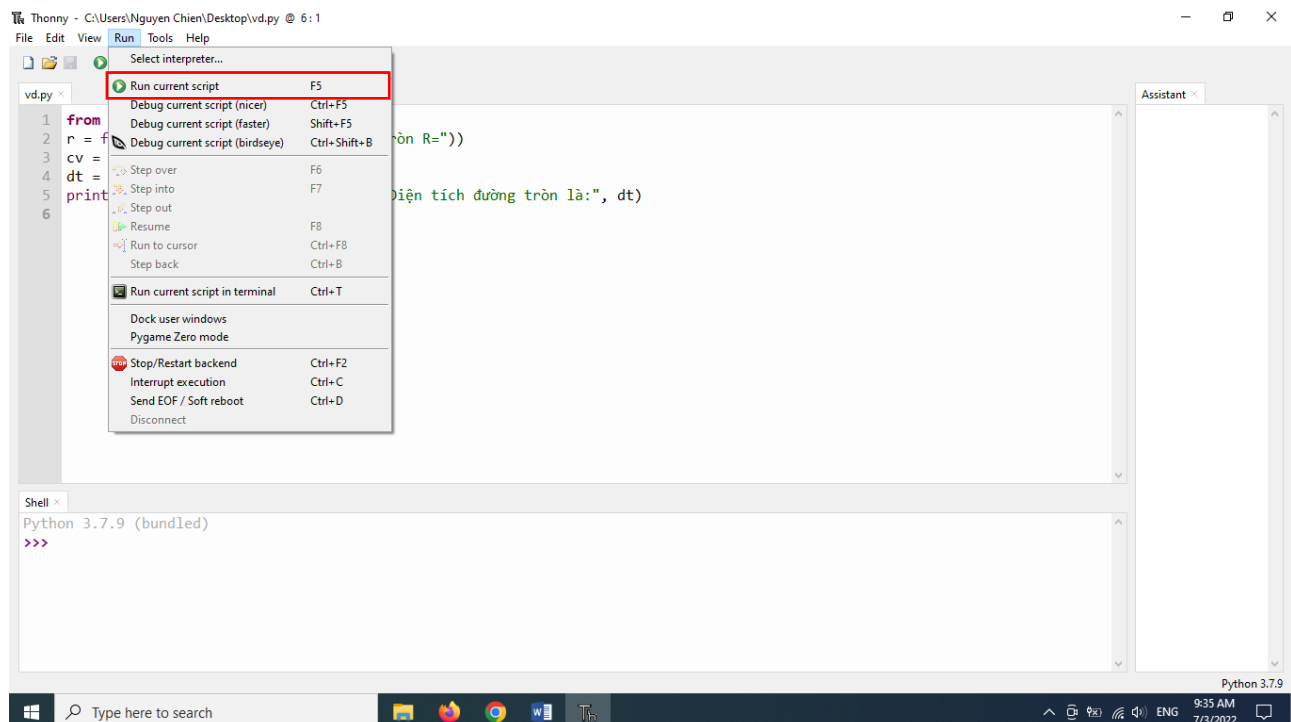
d) Lưu chương trình thành tập tin (File)

Tại chương trình Thony chọn lệnh **File / Save/** Chọn đường dẫn trên đĩa và đặt tên File.



e) Thực thi chương trình và sửa lỗi.

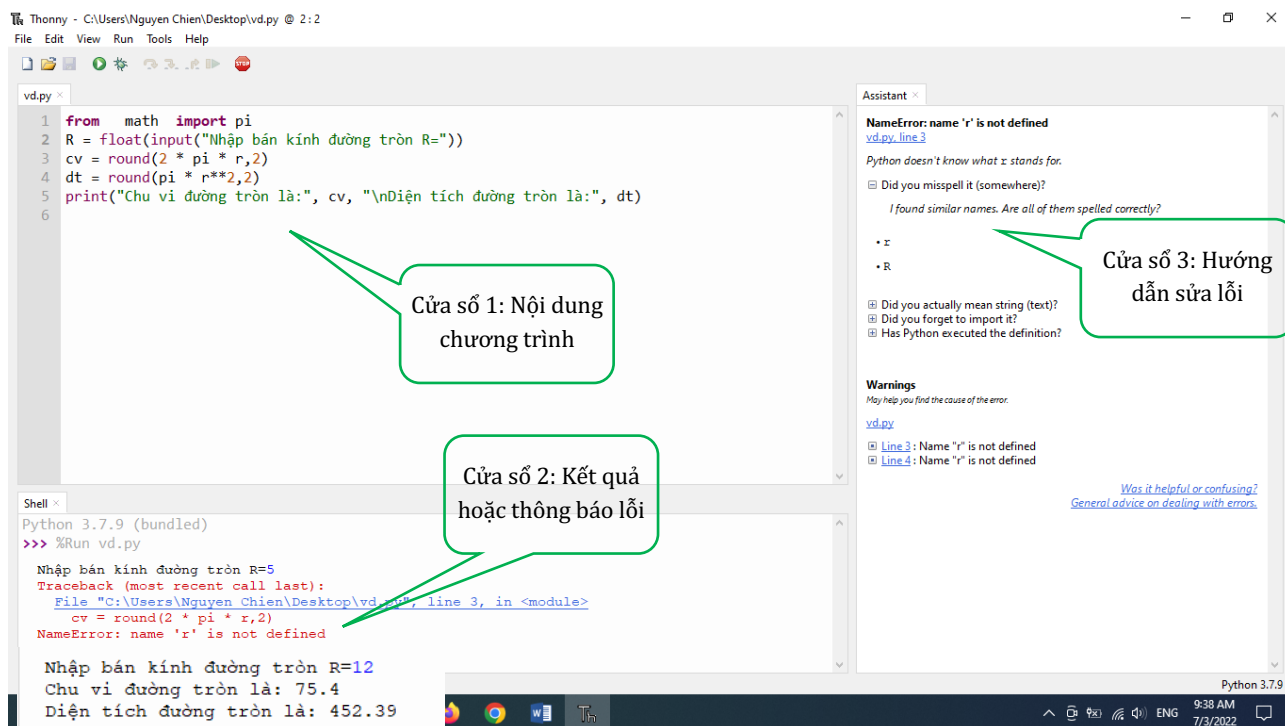
- Để thực thi chương trình, sau khi soạn thảo và lưu chương trình xong người dùng chọn **Menu Run/ Run current script (F5)** hoặc nhấn vào biểu tượng **Run current script** trên thanh công cụ.



- Kết quả thực thi:

Nếu chương trình không có lỗi sẽ được thông dịch và thực thi cho ra kết quả ở cửa sổ số 2.

Nếu chương trình có lỗi sẽ hiển thị thông báo tại cửa sổ số 2 và gợi ý hướng dẫn sửa lỗi tại cửa sổ số 3.



Lưu ý: sau mỗi lần chỉnh sửa chương trình (nếu có) học sinh vào Menu File/ Save để lưu cập nhật những thay đổi vào file trên đĩa.

2. Bài tập thực hành của học sinh.

Học sinh thực hiện viết thuật toán, soạn thảo chương trình trên vở viết và thực hành trên phần mềm cho các bài tập sau:

Câu 1. Viết chương trình thực hiện thao tác nhập dữ liệu từ bàn phím cho a và b, sau đó thực hiện hoán đổi giá trị của a và b rồi in ra màn hình. (**cách 1.** Dùng 3 biến;

cách 2. Dùng 2 biến)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 2. Viết chương trình nhập giá trị cho x và y sau đó thực hiện in giá trị của tổng, tích x và y ra màn hình.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 3. Viết chương trình nhập tuổi của Nam (là một số nguyên dương ≥ 8) từ bàn phím sau đó in ra màn hình tuổi của cha (TCha) và tuổi mẹ (TMe) của Nam, biết tuổi của Cha gấp 4 lần tuổi Nam, tuổi của Mẹ kém Cha là 3 tuổi.

.....

.....

.....

.....

.....

.....

.....

.....

Câu 4. Nhập vào 2 cạnh của một hình chữ nhật. In ra màn hình diện tích và chu vi của nó.

.....

.....

.....

.....

.....

.....

.....

Câu 5. Viết chương trình nhập độ cao h so với mặt đất của một vật rơi tại nơi có gia tốc g , Tính và hiển thị vận tốc của vật tại mặt đất.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 6. Nhập vào thời gian 1 công việc là x giây. Hãy quy đổi thời gian vừa nhập bằng bao nhiêu giờ, bao nhiêu phút, bao nhiêu giây.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 7. Tính x^y với x, y là các số thực nhập từ bàn phím và in kết quả ra màn hình.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 8. Viết chương trình tính diện tích tam giác có 3 cạnh a, b, c nhập từ bàn phím.
 $(p=(a+b+c)/2; \sqrt{p(p-a)(p-b)(p-c)})$

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 9. Viết chương trình nhập từ bàn phím một số nguyên N ($N>100$), hãy in ra màn hình chữ số hàng trăm, hàng chục, hàng đơn vị (ví dụ: số $N=247$, hàng trăm: 2, hàng chục: 4, hàng đơn vị: 7)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 10. Viết chương trình nhập số bông hoa (HOA) từ bàn phím, hãy cho biết có bao nhiêu bó (BO) và bao nhiêu bông lẻ (BL), biết mỗi bó có 12 bông.

.....

.....

.....

.....

.....

.....

.....

.....

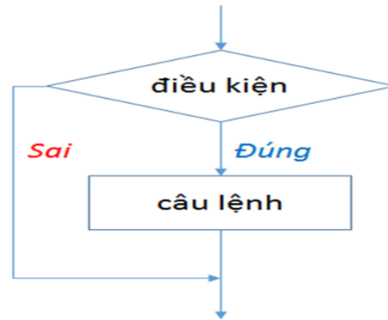
BÀI 6. CẤU TRÚC Rẽ NHÁNH

1. Khái niệm rẽ nhánh.

Rẽ nhánh là cấu trúc lựa chọn khối lệnh tùy theo điều kiện.

Ví dụ: Cho 2 giá trị a và b, nếu a lớn hơn b thì in giá trị a ra màn hình ngược lại in giá trị b ra màn hình.

Như vậy trong ví dụ trên đã có sự lựa chọn giá trị a hoặc giá trị b tùy thuộc vào điều kiện a lớn hơn b hay b lớn hơn a.



2. Câu lệnh if.

- Cú pháp:

if <điều kiện>:

<khối lệnh>

Trong đó:

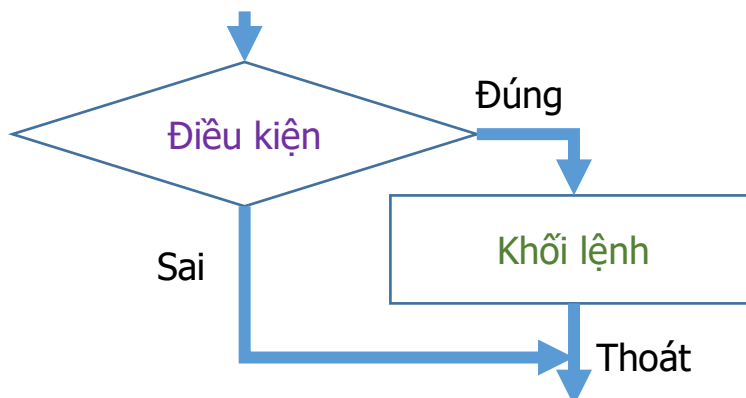
<điều kiện>: là biểu thức logic (trả về giá trị True hoặc False)

<khối lệnh>: là một hoặc nhiều lệnh thực hiện yêu cầu bài toán, phải được viết lùi vào một cấp so với lệnh if.

- Ý nghĩa thực hiện:

Nếu <điều kiện> đúng thì sẽ thực hiện <khối lệnh>

- Sơ đồ thực hiện:



- Ví dụ 1:

```
a = float(input("Nhập a="))
b = float(input("Nhập b="))
if a > b:
    print(a, "lớn hơn", b)
```

Kết quả:

Nhập a = 10

Nhập b = 5

10 lớn hơn 5

- Ví dụ 2:

```
a = float(input("Nhập a="))
b = float(input("Nhập b="))
if a > b:
    print(a, "lớn hơn", b)
    print("a + b =", a+b)
```

Kết quả:

Nhập a = 12

Nhập b = 4

12 lớn hơn 4

12 + 4 = 16

3. Câu lệnh if - else.

- Cú pháp:

if <điều kiện>:

 <khối lệnh 1>

else:

 <khối lệnh 2>

Trong đó:

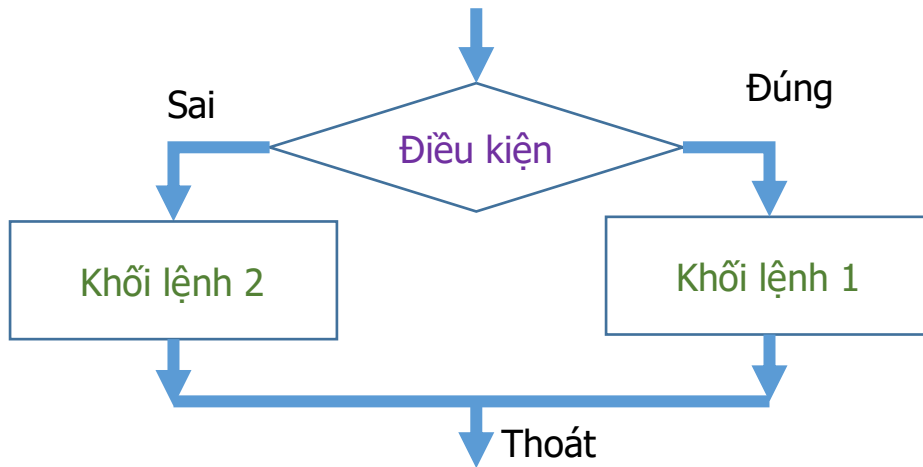
<điều kiện>: là biểu thức logic (trả về giá trị True hoặc False)

<khối lệnh 1>, <khối lệnh 2> : là một hoặc nhiều lệnh thực hiện yêu cầu bài toán, phải được viết lùi vào một cấp so với lệnh if và else.

- Ý nghĩa thực hiện:

Nếu <điều kiện> đúng thì sẽ thực hiện <khối lệnh 1> ngược lại (<điều kiện> sai) thực hiện <khối lệnh 2>

- Sơ đồ thực hiện:



- Ví dụ:

```
a = float(input("Nhập a="))
b = float(input("Nhập b="))
if a > b:
    print(a,"lớn hơn", b)
else:
    print(a,"nhỏ hơn", b)
```

Kết quả:

Nhập a = 8

Nhập b = 12

12 lớn hơn 8

4. Câu lệnh if - elif.

- Cú pháp:

if < điều kiện 1>:

<khối lệnh 1>

elif < điều kiện 2>:

<khối lệnh 2>

Trong đó:

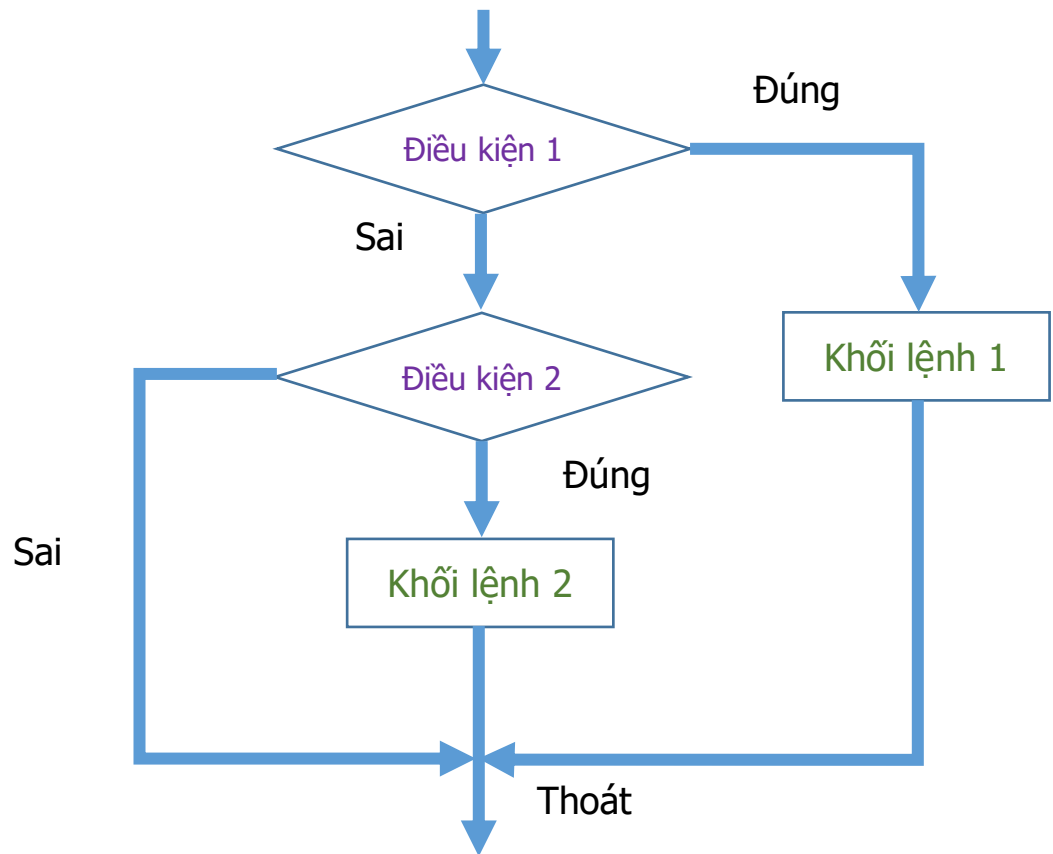
<điều kiện 1>, <điều kiện 2>: là biểu thức logic (trả về giá trị True hoặc False)

<khối lệnh 1>, <khối lệnh 2>: là một hoặc nhiều lệnh thực hiện yêu cầu bài toán, phải được viết lùi vào một cấp so với lệnh if và elif.

- Ý nghĩa thực hiện:

Nếu <điều kiện 1> đúng thì sẽ thực hiện <khối lệnh 1> ngược lại <điều kiện 2> đúng sẽ thực hiện <khối lệnh 2>

- Sơ đồ thực hiện:



- Ví dụ:

```
a = float(input("Nhập a="))
b = float(input("Nhập b="))
if a == b:
    print(a,"bằng", b)
elif a>b:
    print(a,"lớn hơn", b)
elif a<b:
    print(a,"nhỏ hơn", b)
```

Kết quả:

Nhập a = 10

Nhập b = 6

10 lớn hơn 6

Câu hỏi:

Câu 1. Hãy cho biết kết quả của đoạn chương trình sau là gì?

```
a = 20
b = 2
if a == b:
    print("Bạn là thiên tài")
elif a % b == 1:
    z = 30
    print(b**(z-a))
else:
    print(a%b)
```

.....

.....

.....

.....

Câu 2. Viết chương trình nhập giá trị cho a và b, sau đó in ra màn hình số lớn hơn.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 3. Viết chương trình nhập giá trị từ bàn phím giá trị X, cho biết X là số chẵn hay số lẻ.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 4. Viết chương trình nhập giá trị cho 2 số M và N, cho biết M và N có cùng chẵn hoặc cùng lẻ hay không. (VD: M=12, N=4 là cùng chẵn, M=3, N=15 là cùng lẻ)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 5. Viết chương trình nhập độ dài 3 cạnh của một tam giác a, b, c sau đó cho biết tam giác đó có phải tam giác cân hay không. (tự làm thêm trường hợp tam giác đều, tam giác vuông)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 6. Viết chương trình giải phương trình $ax+b=0$, với a và b nhập từ bàn phím.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 7. Nhập vào tâm $O(x_0, y_0)$ và bán kính R của một đường tròn. Sau đó nhập vào một điểm $A(x_a, y_a)$ bất kì và kiểm tra xem nó có thuộc đường tròn hay không?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 8. Giải phương trình bậc 2: $ax^2 + bx + c = 0$, với a, b, c nhập từ bàn phím.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 9. Viết chương trình thu ngân khách sạn:

- Cho người dùng nhập vào một trong 3 loại phòng: A, B hoặc C.
- Người dùng nhập vào số ngày thuê phòng.
- In ra màn hình tiền phải thanh toán:
 - + Nếu loại A thì tiền thành toán là: $300000 * \text{số ngày thuê}$.
 - + Nếu loại B thì tiền thành toán là: $250000 * \text{số ngày thuê}$.
 - + Nếu loại C thì tiền thành toán là: $200000 * \text{số ngày thuê}$.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 10. Viết chương trình tính tiền thuê sách, nhập số cuốn, số ngày, giá. Tính tiền = số ngày * số cuốn * giá. Nếu số ngày >5 thì giảm 2000 ngược lại không giảm. Xuất ra có quy cách 10:2.

.....

.....

.....

.....

.....

.....

.....

.....

BÀI 7. CẤU TRÚC LẶP

1. Khái niệm lặp:

Lặp là thực hiện **nhiều lần một hành động** hoàn toàn giống nhau hoặc chỉ khác nhau về giá trị tính toán.

Ví dụ 1: thực hiện in ra màn hình 100 dòng chữ “chào mừng bạn đến với lập trình Python”. Đây là việc lặp lại nhiều lần hành động giống nhau hoàn toàn.

Ví dụ 2: in ra màn hình các giá trị nguyên trong dãy số [1 ... 100]. Đây là việc lặp lại hành động in ra nhiều lần một con số nhưng giá trị các con số không giống nhau.

2. Cấu trúc lặp For-in

- Cú pháp:

```
for <tên biến> in [tập giá trị]:  
    <khối lệnh>
```

Trong đó:

- ❖ <tên biến>: là một biến kiểu số nguyên (int) dùng để nhận lần lượt các giá trị trong [tập giá trị]
- ❖ [tập giá trị]: là một danh sách các giá trị được sắp xếp theo một trình tự xác định và có một số cách trình bày như sau:
 - Danh sách trực tiếp, các giá trị được viết cách nhau bởi dấu phẩy: 2, 5, 9, 15, 63
 - Bằng hàm **range**(start, end, step)

Biến nhận giá trị theo công thức: **start, start + step, start + 2*step**

Trong đó:

start: là giá trị bắt đầu danh sách. Mặc định là 0 và biến nhận giá trị từ 0 đến end - 1

end: là giá trị xác định giá trị cuối danh sách.

step: là bước nhảy có giá trị mặc định là 1 và luôn khác 0.

VD: **for a in range(2,5):** // bước nhảy là 1

```
    print(a, end=",")
```

Kết quả:

2, 3, 4, 5,

Nếu step > 0 thì start phải nhỏ hơn end và danh sách có giá trị tăng dần bắt đầu từ start.

VD: **for a in range(1, 10, 2):** // bước nhảy là 2

```
print(a, end=",")
```

Kết quả:

1, 3, 5, 7, 9,

Nếu `step < 0` thì `start` phải lớn hơn `end` và danh sách có giá trị giảm dần bắt từ `start`.

VD: **for** `a` **in** `range(10, 1, -2)`: // bước nhảy là -2

```
print(a, end=",")
```

Kết quả:

10, 8, 6, 4, 2,

Lưu ý:

Nếu viết **for** `a` **in** `range(1, 10, -2)`: lệnh sẽ không thực hiện lần nào.

Nếu viết **for** `a` **in** `range(10, 1, 0)`: lệnh sẽ báo lỗi

- Ý nghĩa thực hiện:

Biến sẽ nhận lần lượt các giá trị trong [tập giá trị] với mỗi lần nhận giá trị của biến là một lần <khối lệnh> được thực hiện. Do danh sách giá trị trong [tập giá trị] là hữu hạn và biết trước nên cấu trúc `for-in` còn gọi là cấu trúc lặp với số lần biết trước.

- Ví dụ 1:

```
for a in 5, 14, 23:
    print("python thật thú vị")
```

Kết quả:

python thật thú vị

python thật thú vị

python thật thú vị

- Ví dụ 2:

```
for a in range(10):
    print(a, end=",")
```

Kết quả:

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

- Ví dụ 3:

```
for a in range(1,10,2):
    print(a, end=",")
```

Kết quả:

1, 3, 5, 7, 9,

- Ví dụ 4:

```
for a in range(10,1,-2):  
    print(a, end=",")
```

Kết quả:

10, 8, 6, 4, 2,

- Ví dụ 5:

```
for a in range(10,1,2):  
    print(a)
```

Kết quả:

Lệnh lặp không thực hiện lần nào do $end = 2$ ($end > 0$) nhưng $start$ lại lớn hơn end .

- Ví dụ 6:

```
for a in range(10,1,0):  
    print(a)
```

Kết quả:

Báo lỗi: `ValueError: range() arg 3 must not be zero`

- Ví dụ 7: Viết chương trình giải bài toán tìm kết quả của tích **a*10** mà không dùng phép nhân, với a nhập từ bàn phím và $a > 5$.

- Xác định bài toán:

Input: a, 10.

Output: $P = a * 10$ (không dùng phép nhân)

- Thuật toán:

B1: Nhập a.

B2: $i \leftarrow 0, P \leftarrow 0$.

B3: $P \leftarrow P + a, i \leftarrow i + 1$.

B4: nếu $i < 10$ quay lại B3

B5: in P ra màn hình và kết thúc.

- Viết chương trình

```
p=0
```

```
a= int(input("nhập a= "))
for i in range(0,10):
    p = p+a
print("p = ", p)
```

Kết quả:

nhập a= 5

p = 50

Câu hỏi:

Câu 1. Vì sao cấu trúc lặp For-in được gọi là lặp với số lần biết trước?

.....

.....

.....

.....

.....

Câu 2. Trong cấu trúc lặp for-in, biến đếm hay khối lệnh lần lượt nhận các giá trị trong tập giá trị?

.....

.....

.....

Câu 3. Hãy cho biết ý nghĩa các tham số của hàm range?

.....

.....

.....

.....

.....

Câu 4. Cho lệnh **for a in range(1, 10, -2)**, hãy cho biết biến a sẽ nhận mấy giá trị đó là các giá trị nào và hãy giải thích tại sao biến a nhận giá trị đó?

.....

.....

.....

.....

3. Cấu trúc lặp While

- **Cú pháp:**

while <điều kiện>:
 <khối lệnh>

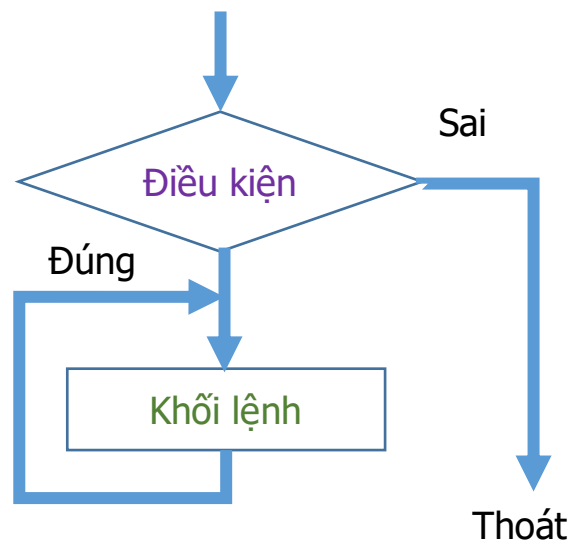
Trong đó:

<điều kiện>: là biểu thức logic

<khối lệnh>: là một hoặc nhiều lệnh sẽ được thực hiện lặp lại nhiều lần.

- **Ý nghĩa thực hiện:**

Cấu trúc while sẽ kiểm tra cho đến khi <điều kiện> còn đúng thì còn thực hiện <khối lệnh>.



Ví dụ 1:

```
a = 1
while a <= 10:
    print(a, end=", ")
    a = a + 1
print("kết thúc lệnh lặp")
```

Kết quả:

1,2,3,4,5,6,7,8,9,10,
kết thúc lệnh lặp

Ví dụ 2: Viết chương trình giải bài toán, cho 2 giá trị nguyên dương a và b ($a > b$), không dùng phép chia hãy cho biết kết quả của a chia cho b được phần nguyên là bao nhiêu và dư bao nhiêu?

- Xác định bài toán
 - Input: a, b
 - Phần nguyên, số dư.

- Thuật toán:

B1. Nhập a, b

B2. Nếu $a \geq b$ thì chuyển B3, ngược lại chuyển B4

B3. $a \leftarrow a - b$

$i \leftarrow i + 1$

Quay lại B2.

B4. In kết quả ra màn hình.

- Viết chương trình

```
a = int(input("nhập a= "))
b = int(input("nhập b= "))
sbc = a
sc = b
i = 0
while a >= b:
    a = a - b
    i = i + 1
print(sbc, "/", sc, "=", i, "dư ", a)
```

Kết quả:

nhập a= 13

nhập b= 4

$13/4 = 3$ dư 1

4. Câu lệnh break

Câu lệnh break được viết trong khối lệnh của lệnh lặp **For-in** hoặc **while**. Khi thực hiện lệnh **break** chương trình sẽ thoát khỏi vòng lặp có chứa lệnh break đó.

Ví dụ :

```
for a in range(1, 6):
```

```
if a == 3:
    break
print(a)
```

Kết quả:

1

2

5. Câu lệnh continue

Câu lệnh continue được viết trong khối lệnh của vòng lặp. Khi thực hiện lệnh **continue** các lệnh phía sau trong khối lệnh sẽ không được thực hiện mà sẽ tiếp tục lần lặp tiếp theo.

Ví dụ :

```
for a in range(1,10):
    if a % 2 == 0:
        continue
    print(a, "là số lẻ")
```

Kết quả:

1 là số lẻ

3 là số lẻ

5 là số lẻ

7 là số lẻ

9 là số lẻ

Câu hỏi:

Câu 1. Viết chương trình in ra các số lẻ nhỏ hơn hoặc bằng n (Với n>100 được nhập từ bàn phím).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 2. Viết chương trình in ra các số chẵn nhỏ hơn hoặc bằng n (Với n được nhập từ bàn phím). Sao cho 15 số được in trên một dòng.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 3. Viết chương trình in ra tổng các số lẻ nhỏ hơn hoặc bằng n (Với n được nhập từ bàn phím).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 4. In bảng cửu chương n (2-9). Với n nhập từ bàn phím.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 5. Viết chương trình xét xem một số n có phải là số nguyên tố không?

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Câu 6. Viết chương trình tính giai thừa của N (với $N > 4$ nguyên dương nhập từ bàn phím).

.....

.....

.....

.....

.....

.....

.....

.....

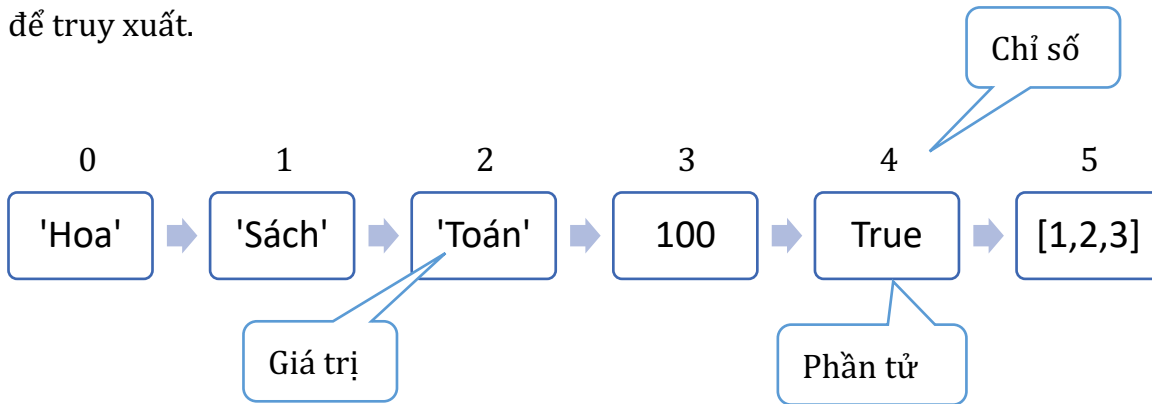
.....

.....

BÀI 8. KIỂU DỮ LIỆU DANH SÁCH - LIST

1. Khái niệm:

Danh sách (list) là kiểu dữ liệu mà mỗi biến có thể chứa một tập hợp các phần tử, các phần tử có thể có giá trị khác nhau về kiểu dữ liệu, mỗi phần tử được gán một chỉ số để truy xuất.



2. Khái báo biến danh sách.

- **Cú pháp 1:** Khai báo danh sách cùng với giá trị các phần tử.

$\langle \text{biến danh sách} \rangle = [\langle \text{phần tử } 0 \rangle, \langle \text{phần tử } 1 \rangle, \langle \text{phần tử } 2 \rangle, \dots, \langle \text{phần tử } n-1 \rangle]$

Lưu ý:

+ Các phần tử được liệt kê trong cặp dấu ngoặc vuông [], viết cách nhau bởi dấu phẩy.

+ Danh sách các phần tử không cố định mà có thể thay đổi tùy ý.

+ Số lượng phần tử trong danh sách còn gọi là độ dài ký hiệu là n.

- **Ví dụ 1:**

color = ['Đỏ', 'Cam', 'Vàng', 'Lục', 'Lam', 'Tràm', 'Tím']

- **Cú pháp 2:** Khai báo danh sách rỗng (không có phần tử).

$\langle \text{biến danh sách} \rangle = []$

Hoặc

$\langle \text{biến danh sách} \rangle = \text{list}()$

- **Ví dụ 2:**

hocsinh = []

monhoc = list()

- **Cú pháp 3:** Khai báo danh sách gồm n phần tử đều có giá trị rỗng.

$\langle \text{biến danh sách} \rangle = [\text{None}] * n$

- Ví dụ 3:

hocsinh = []*45

- Cú pháp 4: tạo danh sách từ một chuỗi cho trước.

<biến danh sách> = list(<chuỗi>)

- Ví dụ:

```
xau = 'hoclaptrinh'
ds = list(xau)
print(ds)
```

Kết quả:

['h', 'o', 'c', 'l', 'a', 'p', 't', 'r', 'i', 'n', 'h']

3. Truy cập đến các phần tử trong danh sách.

- Cú pháp:

<biến danh sách>[<chỉ số>]

Trong đó:

Chỉ số được đánh bắt đầu 0, 1, 2, ... (từ trái sang phải) và bắt đầu -1, -2, ... (từ phải sang trái)

Chỉ được truy cập đến chỉ số trong phạm vi hiện có của biến danh sách.

- Ví dụ:

```
color = ['Đỏ', 'Cam', 'Vàng', 'Lục', 'Lam', 'Tràm', 'Tím']
print(color[3])
print(color[-2])
print(color[7])
```

Kết quả:

Lục

Tràm

IndexError: list index out of range

4. Nhập, in giá trị phần tử của biến danh sách.

Để nhập giá trị cho từng phần tử của biến danh sách chúng ta kết hợp phép gán và cấu trúc lặp.

Để in các giá trị của biến danh sách chúng ta sử dụng hàm print.

Ví dụ:

```
mylist=[None]*5
for i in range(5):
    val=float(input('Nhập giá trị ' + str(i) + ':'))
    mylist[i] = val
print(mylist)
```

Kết quả:

Nhập giá trị 0: 1

Nhập giá trị 1: 3

Nhập giá trị 2: 6

Nhập giá trị 3: 8

Nhập giá trị 4: 9

[1.0, 3.0, 6.0, 8.0, 9.0]

5. Các phương thức trên danh sách.

a. Lấy số lượng phần tử của danh sách.

- Cú pháp:

len(<biến danh sách>)

- Ví dụ:

```
color = ['Đỏ', 'Cam', 'Vàng', 'Lục', 'Lam', 'Tràm', 'Tím']
Print('Số phần tử:', len(color))
```

Kết quả:

Số phần tử: 7

b. Nối danh sách.

- Cú pháp:

<biến danh sách 1> + <biến danh sách 2> + ... + <biến danh sách n>

Trả về là một danh sách gồm các phần tử của các danh sách thành phần theo đúng trình tự của các danh sách đem ra nối.

- Ví dụ:

```
color = ['Đỏ', 'Cam', 'Vàng', 'Lục', 'Lam', 'Tràm', 'Tím']
monhoc = ['Toán', 'Lý', 'Hóa']
hoa = ['Đào', 'Mận', 'Hồng', 'Cúc']
tonghop = hoa + color + monhoc
```

```
print(tonghop)
```

Kết quả:

```
['Đào', 'Mận', 'Hồng', 'Cúc', 'Đỏ', 'Cam', 'Vàng', 'Lục',  
'Lam', 'Tràm', 'Tím', 'Toán', 'Lý', 'Hóa']
```

c. Lặp danh sách.

- Cú pháp:

<biến danh sách> * <số lần lặp> hoặc <số lần lặp> <biến danh sách>

Trả về một danh sách bằng cách ghép chính danh sách đó <số lần lặp>.

- Ví dụ:

```
hoa = ['Đào', 'Mận', 'Hồng', 'Cúc']  
laphoa = hoa * 3  
print(laphoa)
```

Kết quả:

```
['Đào', 'Mận', 'Hồng', 'Cúc', 'Đào', 'Mận', 'Hồng',  
'Cúc', 'Đào', 'Mận', 'Hồng', 'Cúc']
```

d. Thêm phần tử vào cuối danh sách.

- Cú pháp:

<biến danh sách> . append(<giá trị cần thêm>)

- Ví dụ:

```
hoa = ['Đào', 'Mận', 'Hồng', 'Cúc']  
hoa.append(5)  
print(hoa)
```

Kết quả:

```
['Đào', 'Mận', 'Hồng', 'Cúc', 5]
```

e. Chèn phần tử vào danh sách.

Cú pháp: <biến danh sách>.insert(i, x)

Chèn phần tử x vào vị trí có chỉ số i trong danh sách

Ví dụ:

```
hoa = ['Đào', 'Mận', 'Hồng', 'Cúc']  
hoa.insert(3, 'Mai')  
print(hoa)
```

Kết quả:

```
[ 'Đào', 'Mận', 'Hồng', 'Mai', 'Cúc' ]
```

f. Hàm xóa phần tử khỏi danh sách.

Phương thức remove()

Cú pháp: <biến danh sách>.remove (x)

Xóa phần tử x xuất hiện đầu tiên trong danh sách. Nếu x không có trong danh sách sẽ báo lỗi.

Ví dụ:

```
num = [100, 5, 12, 5, 3, 5]
num.remove(5)
print(num)
```

Kết quả:

```
[100, 12, 5, 3, 5]
```

Hàm del

Cú pháp: del <biến danh sách>[index]

Xóa phần tử có chỉ số index trong danh sách.

Ví dụ:

```
color = ['Đỏ', 'Cam', 'Vàng', 'Lục', 'Lam', 'Tràm', 'Tím']
del color[4]
print(color)
```

Kết quả:

```
[ 'Đỏ', 'Cam', 'Vàng', 'Lục', 'Tràm', 'Tím' ]
```

g. Phép toán in và not in

Phép toán: <phần tử> **in** <biến danh sách>

Trả về giá trị **True** nếu <phần tử> thuộc <biến danh sách> ngược lại trả về giá trị **False**.

Ví dụ:

```
color = ['Đỏ', 'Cam', 'Vàng', 'Lục', 'Lam', 'Tràm', 'Tím']
x = 'Lam' in color
print(x)
```

Kết quả:

True

Phép toán: <phần tử> **not in** <biến danh sách>

Trả về giá trị **True** nếu <phần tử> không thuộc <biến danh sách> ngược lại trả về giá trị **False**.

Ví dụ:

```
color = ['Đỏ', 'Cam', 'Vàng', 'Lục', 'Lam', 'Tràm', 'Tím']
x = 'Lam' not in color
print(x)
```

Kết quả:

False

h. Phương thức index.

Cú pháp:

<biến danh sách> . **index**(<giá trị>)

Trả về chỉ số của phần tử đầu tiên trong sách bằng <giá trị>, ngược lại trả về lỗi **ValueError: 'giá trị' is not in list**

Ví dụ:

```
color = ['Đỏ', 'Cam', 'Vàng', 'Lục', 'Lam', 'Tràm', 'Tím']
x = color.index('Lam')
y = color.index('a')
print(x)
print(y)
```

Kết quả:

4

ValueError: 'a' is not in list

i. Phép toán max, min, sum trên danh sách.

Phép toán **min**(<biến danh sách>) trả về giá trị của phần tử nhỏ nhất trong <biến danh sách>

Phép toán **max**(<biến danh sách>) trả về giá trị của phần tử lớn nhất trong <biến danh sách>

Phép toán **sum**(<biến danh sách>) trả về giá trị là tổng các giá trị trong <biến danh sách>

Ví dụ:

```
num = [100, 7, 12, 5, 3, 19]
print('Giá trị nhỏ nhất:', min(num))
print('Giá trị lớn nhất:', max(num))
print('Tổng giá trị:', sum(num))
```

kết quả:

Giá trị nhỏ nhất: 3

Giá trị lớn nhất: 100

Tổng giá trị: 146

j. Đếm số lần xuất hiện của một phần tử có trong danh sách.

- **Cú pháp:** <biến danh sách>.count(x)

Đếm số lần xuất hiện của một phần tử x trong danh sách.

- **Ví dụ:**

```
color = ['Đỏ', 'Cam', 'Vàng', 'Lục', 'Lam', 'Tràm', 'Vàng']
num = [100, 5, 12, 5, 3, 5]
x = 'Vàng'
y = 5
print('Số lần xuất hiện của x:', color.count(x))
print('Số lần xuất hiện của y:', num.count(y))
```

Kết quả:

Số lần xuất hiện của x: 2

Số lần xuất hiện của y: 3

k. Sắp xếp danh sách.

- Sắp xếp tăng dần: <biến danh sách>.sort()

- Sắp xếp giảm dần: <biến danh sách>.sort(reverse = True)

Ví dụ:

```
num = [100, 7, 12, 5, 3, 19]
num.sort( )
print(num)
```

Kết quả:

[3, 5, 7, 12, 19, 100]

I. Phương thức đảo ngược danh sách.

Cú pháp: <biến danh sách>.reverse()

Đảo ngược thứ tự các phần tử trong danh sách.

Ví dụ:

```
num = [100, 7, 12, 5, 3, 19]
num.reverse( )
print(num)
```

Kết quả:

[19, 3, 5, 12, 7, 100]

Câu hỏi

Cho 1 dãy số gồm n phần tử nhập ngẫu nhiên ($n \geq 5$). Giá trị từng phần tử là số nguyên. Thực hiện các chương trình sau:

Câu 1. Nhập từ bàn phím số phần tử và giá trị từng phần tử.

.....

.....

.....

.....

.....

.....

Câu 2. Tính tổng các phần tử âm của dãy

.....

.....

.....

.....

.....

.....

Câu 3. Tính tổng và trung bình cộng của các phần tử chẵn của dãy

.....

.....

.....

.....

Câu 4. Tìm phần tử có giá trị lớn nhất của dãy

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

BÀI 9. KIỂU DỮ LIỆU XÂU

1. Khái niệm.

Trong Python chuỗi là một dãy ký tự, được đặt trong cặp nháy kép (" ") hoặc cặp nháy đơn (' '), chuỗi rỗng là chuỗi không có ký tự nào.

Chuỗi trong Python có hỗ trợ tiếng Việt.

Ký tự trong chuỗi được đánh chỉ số để quản lý, bắt đầu từ trái sang phải là 0,1,..., từ phải sang trái là -1, -2, ...

Số ký tự trong chuỗi được gọi là độ dài.

Trong Python khi ta gán một chuỗi cho biến thì tự động biến có kiểu dữ liệu chuỗi.

Khoảng trắng (dấu cách) cũng được coi là một ký tự trong chuỗi.

Xâu	P	y	t	h	o	n
Chỉ số	0	1	2	3	4	5
	-6	-5	-4	-3	-2	-1

Ví dụ:

hoten = "Đỗ Tiến Sĩ"

bidanh = 'Giáo sư'

2. Thao tác với chuỗi.

a. Hàm lấy độ dài.

Cú pháp: **len**(<biến chuỗi>)

Ví dụ:

```
hoten = 'Đỗ Tiến Sĩ'
print('Độ dài tên:', len(hoten))
```

Kết quả:

Độ dài tên:10

b. Truy cập đến ký tự trong chuỗi.

Cú pháp: <biến chuỗi>[<chỉ số>]

Ví dụ:

```
hoten = 'Đỗ Cử Nhân'
print(hoten[3])
print(hoten[len(hoten)-1])
```

Kết quả:

C

n

c. Ghép xâu.

Cú pháp:

<xâu 1> + <xâu 2> + ... +<xâu n>

Ví dụ:

```
ho = 'Lệnh Hồ'  
ten = 'Sung'  
bidanh = 'Bang Chủ'  
print( bidanh + ' ' + ho + ' ' + ten)
```

Kết quả:

Bang Chủ Lệnh Hồ Sung

d. Lặp xâu.

Cú pháp:

<biến xâu> * <Số lần lặp>

<Số lần lặp> * <biến xâu>

Ví dụ:

```
hoten = 'Đỗ Tiên Sĩ'  
lapxau3 = hoten*3  
lapxau5 = 5* hoten  
print(lapxau3)  
print(lapxau5)
```

Kết quả

Đỗ Tiên SĩĐỗ Tiên SĩĐỗ Tiên Sĩ

Đỗ Tiên SĩĐỗ Tiên SĩĐỗ Tiên SĩĐỗ Tiên SĩĐỗ Tiên Sĩ

e. So sánh hai xâu.

So sánh 2 xâu được thực hiện bằng cách so sánh lần lượt mã ký tự của từng cặp ký tự từ trái sang phải, nếu ký tự có mã lớn hơn thì kết luận xâu đó là xâu lớn hơn.

Ví dụ: 'abe' > 'acd' -> False

```
a = 'abe'  
b = 'acd'
```

```
if a>b:
    print('a>b')
else:
    print('b>a')
```

Kết quả

b>a

f. Lấy xâu con.

Cú pháp: <biến xâu>[a:b]

Dùng để lấy xâu con từ <biến xâu> gồm các ký tự liên tiếp từ chỉ số **a** đến chỉ số **b-1** (với điều kiện $a \leq b$)

Ví dụ:

```
hoten = 'Đỗ Tiến Sĩ'
holot = hoten[3:7]
ten = hoten[8:10]
print(holot)
print(ten)
```

kết quả:

Tiến

Sĩ

g. Đảo ngược xâu.

Cú pháp: <biến xâu>[::-1]

Dùng để đảo ngược vị trí các ký tự trong xâu.

Ví dụ:

```
hoten = 'ambulance'
print('Xâu đảo ngược:', hoten[::-1])
```

Kết quả:

Xâu đảo ngược: ecnalubma

h. Chuyển đổi ký tự hoa, thường, hoa đầu từ, đếm ký tự.

Phương thức: <biến xâu>.upper()

Chuyển đổi các ký tự trong <biến xâu> thành ký tự in hoa.

Phương thức: <biến xâu>.lower()

Chuyển đổi các ký tự trong <biến xâu> thành ký tự thường.

Phương thức: **<biến xâu>.capitalize()**

Chuyển đổi ký tự đầu tiên của <biến xâu> thành ký tự in hoa.

Phương thức: **<biến xâu>.title()**

Chuyển đổi ký tự đầu tiên mỗi từ của <biến xâu> thành ký tự in hoa.

Phương thức: **<biến xâu>.count(<xâu con>)**

Đếm số lần xuất hiện của <xâu con> trong <biến xâu>.

Lưu ý: các phương thức không làm thay đổi nội dung <biến xâu>

Ví dụ:

```
x1 = 'học lập trình'
x2 = 'Học Lập Trình'
x3 = 'h'

print('Xâu in hoa:', x1.upper())
print('Xâu thường:', x2.lower())
print('In hoa đầu xâu:', x1.capitalize())
print('In hoa đầu mỗi từ:', x1.title())
print('Số lần xuất hiện ký tự', x3, 'là:', x1.count(x3))
```

Kết quả:

```
Xâu in hoa: HỌC LẬP TRÌNH
Xâu thường: học lập trình
In hoa đầu xâu: Học lập trình
In hoa đầu mỗi từ: Học Lập Trình
Số lần xuất hiện ký tự h là: 2
```

i. Phép toán in.

Cú pháp: <xâu 1> **in** <xâu 2>

Kiểm tra <xâu 1> có tồn tại bên trong <xâu 2> hay không. Nếu có tồn tại sẽ trả về giá trị **True** ngược lại trả về giá trị **False**

Ví dụ:

```
x2 = 'Học lập trình'
x1 = 'lập'
if x1 in x2:
```

```
print(x1,'là xâu con của', x2)
else:
    print(x2,'không chứa', x1)
```

Kết quả:

lập là xâu con của Học lập Trình

Câu hỏi: viết chương trình thực hiện các bài toán sau.

Câu 1. Nhập xâu kí tự bất kì

a. Đếm số lần xuất hiện của 1 kí tự do người dùng nhập trong xâu.

Ví dụ: nghieng -> g- xuất hiện 2 lần

b. Liệt kê các kí tự có mặt trong xâu cùng số lần xuất hiện của các kí tự đó.

Ví dụ: DFGJGKGH; Số lần xuất hiện của các ký tự: D = 1; G = 3; F = 2; J = 2; K = 1; H = 1

.....

.....

.....

.....

.....

.....

.....

.....

Câu 2. Nhập 1 xâu kí tự. Xét xem trong xâu có K kí tự kề nhau mà giống nhau hay không?

.....

.....

.....

.....

.....

.....

.....

.....

Câu 3. Nhập 1 xâu kí tự. Kiểm tra tính đối xứng của xâu đó. Nếu xâu không đối xứng thì đảo xâu.

Ví dụ: Xâu đối xứng abcdedcba -> không đảo

Xâu thường: sbhtr -> rthbs

.....

.....

.....

.....

.....

.....

.....

Câu 4. Nhập vào họ tên bất kì sau đó biến đổi các chữ cái đầu tiên của từ là in hoa.

Ví dụ: htrh trjtrj rthjtrjty -> Htrh Trjtrj Rthjtrjty

.....

.....

.....

.....

.....

.....

.....

BÀI 10. KIỂU DỮ LIỆU TẬP

1. Mở tệp để ghi dữ liệu.

a. Khai báo biến tệp

Cú pháp: <biến tệp> = **open**(<tên tệp>, 'w')

Trong đó:

- <tên tệp> là tên tệp tin lưu trên đĩa hoặc gồm cả đường dẫn và tên tệp tin.
- w có nghĩa là ghi vào tệp tin.

Khi thực hiện lệnh trên nếu tệp chưa tồn tại lệnh sẽ tạo một tệp mới với nội dung trống, nếu tệp đã tồn tại thì lệnh sẽ mở tệp lên, trong trường hợp này người dùng ghi dữ liệu mới vào thì dữ liệu đang có trong tệp sẽ bị xóa trắng.

Ví dụ:

```
hoso = open('lylich.txt', 'w')
ds = open('d:/11B5tin/hocsinh.txt', 'w')
```

b. Ghi dữ liệu vào tệp.

Cú pháp: <biến tệp>.**write**(<xâu ký tự>)

Lệnh dùng để ghi một xâu ký tự vào nội dung của tệp.

Ví dụ:

```
ds = open('d:/hocsinh.txt', 'w')
ds.write('Ho va ten: Hoang Duoc Su\n')
ds.write('Noi o: Dao Dao hoa\n')
ds.write('Bi danh: Hoang Lao Ta')
```

2. Mở tệp để đọc dữ liệu.

a. Khai báo biến tệp.

Cú pháp: <biến tệp> = **open**(<tên tệp>, 'r')

Khi thực hiện lệnh tệp có <tên tệp> sẽ được mở và gán cho <biến tệp>, nếu <tên tệp> không tồn tại chương trình sẽ báo lỗi **FileNotFoundError**.

Ví dụ:

```
docds = open('D:/hocsinh.txt', 'r')
```

b. Đọc dữ liệu.

Cú pháp: <biến tệp>.**read**()

Lệnh sẽ trả về tất cả các ký tự trong tệp kể cả ký tự xuống dòng và lưu thành một chuỗi ký tự.

Ví dụ:

```
docds = open('D:/hocsinh.txt', 'r')
noidung = docds.read()
print(noidung)
```

3. Mở tệp để ghi thêm dữ liệu.

Cú pháp: <biến tệp> = **open**(<tên tệp>, 'a')

Lệnh thực hiện sẽ cho phép thực hiện lệnh ghi thêm dữ liệu và cuối tệp mà không làm mất dữ liệu đã có trước đó của tệp.

Ví dụ:

```
ds = open('d:/hocsinh.txt', 'a')
ds.write('Con gái: Hoang Dung\n')
```

4. Đóng tệp

Cú pháp: <biến tệp>.**close**()

Lệnh sẽ thực hiện việc đóng tệp sau khi thao tác xong với tệp vì vậy lệnh này thường đặt ở cuối chương trình.

Ví dụ:

```
ds = open('d:/hocsinh.txt', 'a')
ds.write('\nDe tu: Than Dieu Dai Hiep')
ds.close()
```

5. Câu hỏi.

Học sinh thực hiện khai báo một tệp với tên là **hocsinh.txt** lưu trên ổ đĩa D và thực hiện các thao tác sau:

- Mở tệp để ghi các thông tin (mỗi thông tin trên một dòng) của học sinh vào tệp gồm:

- + Họ và tên
- + Ngày sinh
- + Lớp
- + Nơi sinh
- + Giới tính.

- Đọc dữ liệu từ tệp và in ra màn hình.

- Ghi thêm thông tin vào tệp gồm mỗi thông tin trên một dòng:

+ Địa chỉ nơi ở.

+ Họ tên Cha.

+ Họ tên mẹ.

+ Quê quán.

- Đóng tệp.

[illegible]

BÀI 11. CHƯƠNG TRÌNH CON

1. Khái niệm.

Chương trình con là một dãy các lệnh mô tả một số thao tác thực hiện một công việc là bộ phận của một công việc lớn hơn và có thể được gọi từ nhiều vị trí trong chương trình.

Vị trí khai báo chương trình con có thể là bất kỳ vị trí nào trong chương trình chỉ cần trước lệnh gọi chương trình con.

Ví dụ: việc tính Delta trong bài toán giải phương trình bậc 2, thì đoạn lệnh tính Delta được coi là một chương trình con.

2. Cú pháp xây dựng chương trình con.

a. Chương trình con không trả về giá trị.

def <tên chương trình con>(<tham số>):
 [<khối lệnh>]

Trong đó:

- Tên chương trình con: đặt theo quy tắc đặt tên, ngắn gọn, gợi nhớ chức năng của chương trình.
- Tham số: là các giá trị được người dùng đưa vào để tham gia tính toán bên trong chương trình con. Lưu ý chương trình con có thể không có tham số.
- Khối lệnh là dãy lệnh thực hiện chức năng của chương trình con.

Ví dụ 1: viết chương trình con in hình cây thông.

```
def caythong( ):
    print('    *')
    print('   ***')
    print('  *****')
    print(' *****')
    print('    *')
    print('    *')
    print('    *')
```

b. Chương trình con trả về giá trị.

def <tên chương trình con>(<tham số>):
 [<khối lệnh>]
 return <giá trị>

Trong đó:

- Tên chương trình con: đặt theo quy tắc đặt tên, ngắn gọn, gợi nhớ chức năng của chương trình.
- Tham số: là các giá trị được người dùng đưa vào để tham gia tính toán bên trong chương trình con. Lưu ý chương trình con có thể không có tham số.
- Khối lệnh là dãy lệnh thực hiện chức năng của chương trình con.
- Giá trị là kết quả trả về sau khi thực hiện xong chương trình con.

Ví dụ 2: viết chương trình con tính Delta.

```
def    tinhdelta(a, b, c):  
    delta = b**2 - 4*a*c  
    return delta
```

c. Chương trình con dùng tham số mặc định.

def <tên chương trình con>(<tham số>= <giá trị mặc định>):

Khi gọi chương trình con nếu người dùng có truyền tham số thì chương trình sẽ sử dụng tham số đó để tính toán trong chương trình, ngược lại người dùng không đưa tham số khi gọi chương trình con thì chương trình lấy giá trị mặc định để tính toán.

Ví dụ 3: in số lượng cây thông tùy chọn.

```
def    caythongtuychon(n=1):  
    for i in range(n):  
        print('    *')  
        print('    ***')  
        print('    *****')  
        print('    *****')  
        print('    *')  
        print('    *')  
        print('    *')
```

3. Sử dụng chương trình con.

Chương trình con được sử dụng (gọi) bất kỳ vị trí nào trong chương trình chỉ cần đáp ứng điều kiện phía sau phần khai báo chương trình con.

Cú pháp: <tên chương trình con>(<tham số>)

Trong đó:

- Tên chương trình con: gọi theo tên đã được khai báo.

- Tham số (nếu có): đưa giá trị vào tham gia tính toán trong chương trình con.

Ví dụ 4: gọi chương trình con in một cây thông.

```
caythong()
```

kết quả:

```

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

```

Ví dụ 5: gọi chương trình con in số cây thông tùy chọn.

```
caythong(3)
```

kết quả:

```

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

```

Ví dụ 6: gọi chương trình con tính Delta.

```
Print('Delta=', tinhdelta(5, 3, -6))
```

kết quả:

$\Delta = 129$

4. Biến toàn cục và biến cục bộ.

[illegible]

This image shows a full page of white paper with horizontal dashed lines. The lines are evenly spaced and run across the width of the page, providing a guide for writing. There are no margins, text, or other markings on the page.

