

# CHƯƠNG TRÌNH CON VÀ THƯ VIỆN CÁC CHƯƠNG TRÌNH CON CÓ SẴN

**Học xong bài này, em sẽ:**

- ✓ Xây dựng và sử dụng được chương trình con trong Python.
- ✓ Sử dụng được chương trình con xây dựng sẵn của hệ thống.



Khi giải quyết một bài toán phức tạp, ta có thể phân chia nó thành một số bài toán con. Trong lập trình có khái niệm chương trình con, em hãy đoán xem chương trình con của một chương trình là gì?

## 1 Khái niệm chương trình con



1

Khi giải quyết một bài toán phức tạp, người ta thường phân chia bài toán đó thành một số bài toán con. Em sẽ chia bài toán sau đây thành những bài toán con nào?

*Bài toán:* Cho ba tam giác có độ dài ba cạnh lần lượt là a, b và c; u, v và w; p, q và r. Độ dài các cạnh đều là số thực cùng đơn vị đo. Em hãy tính diện tích của mỗi tam giác đó và đưa ra diện tích lớn nhất trong các diện tích tính được. Công thức Heron tính diện tích tam giác theo độ dài ba cạnh:

$$S = \frac{\sqrt{(a+b+c)(a+b-c)(a-b+c)(b+c-a)}}{4}$$

Khi lập trình để giải một bài toán có thể chia bài toán đó thành các bài toán con, viết các đoạn chương trình giải các bài toán con. Sau đó xây dựng chương trình giải quyết bài toán ban đầu bằng cách sử dụng các đoạn chương trình đã viết cho các bài toán con.

Các ngôn ngữ lập trình bậc cao đều cho phép người lập trình tạo ra chương trình con bằng cách đặt tên cho một đoạn chương trình gồm các câu lệnh thực hiện một việc nào đó. Mỗi khi cần thực hiện việc này sẽ không cần viết lại các câu lệnh này mà chỉ viết ra tên đã đặt cho đoạn chương trình. Đoạn chương trình được đặt tên như thế chính là một chương trình con và tên đã đặt là tên của chương trình con đó. Sử dụng các chương trình con là một trong những cách giúp việc lập trình trở nên dễ dàng hơn (minh họa ở *Hình 1*).

```

File Edit Format Run Options Window Help
def Hello():
    hoten = input('Xin cho biết tên bạn? ')
    chaomung = ' Xin chào bạn ' + hoten
    print(chaomung)

Hello()

```

Khối lệnh được đặt tên Hello  
Gọi tên hàm ở những nơi cần thực hiện

Hình 1. Một chương trình Python có chương trình con Hello

## ② Khai báo và gọi thực hiện một hàm trong Python

Có thể gọi một chương trình con trong Python là một hàm. Để sử dụng hàm cần khai báo hàm và viết lời gọi thực hiện. Hàm trong Python được khai báo theo mẫu sau:

```
def tên_hàm(tham số):
    Các lệnh mô tả hàm
```

Trong đó:

- Tên hàm phải theo quy tắc đặt tên trong Python.
- Theo sau tên hàm có thể có hoặc không có các tham số.
- Phần thân hàm (gồm các lệnh mô tả hàm) phải viết lùi vào theo quy định của Python.

Ví dụ:

<i>Tên hàm</i>	<i>Danh sách tham số rỗng</i>
<i>Từ khóa khai báo hàm</i>	<pre> File Edit Format Run Options Window Help def ptb1(): # Giải phương trình bậc nhất     a = int(input("a= "))     b = int(input("b= "))     if a != 0:         print("Phương trình có nghiệm duy nhất: ", -b/a)     elif b == 0:         print(' Phương trình có vô số nghiệm.')     else:         print(' Phương trình vô nghiệm.') </pre> <p>ptb1() ← Lời gọi thực hiện hàm</p>

*Các lệnh mô tả hàm*

Hình 2. Một chương trình Python có sử dụng hàm

```

File Edit Shell Debug Options Window Help
a = 5
b = 4
Phương trình có nghiệm duy nhất: -0.8
>>>

```

```

File Edit Shell Debug Options Window Help
a = 0
b = 0
Phương trình có vô số nghiệm.
>>>

```

```

File Edit Shell Debug Options Window Help
a = 0
b = 4
Phương trình vô nghiệm.
>>>

```

Hình 3. Một số kết quả chạy chương trình ở Hình 2

### 3 Chuyển dữ liệu cho hàm thực hiện



2

Chương trình trong *Hình 2* khai báo hàm `ptb1()`, hàm này giải phương trình có dạng  $ax + b = 0$ . Khi được gọi thực hiện, hàm `ptb1()` yêu cầu nhập các hệ số  $a$ ,  $b$  từ bàn phím, biện luận và giải phương trình rồi đưa ra kết quả.

- 1) Em hãy soạn thảo chương trình ở *Hình 2* đặt tên là “`VD_ptb1.py`”, sau đó chạy chương trình với các dữ liệu đầu vào như ở *Hình 3* và đối chiếu kết quả.
- 2) Em hãy sửa lại chương trình “`VD_ptb1.py`” theo các bước trong *Bảng 1*, đặt tên là “`Try_ptb1.py`”, chạy thử và trả lời hai câu hỏi sau:
  - a) Chương trình “`Try_ptb1.py`” đã truyền trực tiếp hệ số  $a = 5$ ,  $b = 4$  vào lời gọi hàm `ptb1(5, 4)`, kết quả khi chạy có khác gì với kết quả chạy chương trình ở *Hình 2* không?
  - b) Vì sao trong chương trình “`Try_ptb1.py`”, thân của hàm không cần những câu lệnh nhập giá trị cho các hệ số  $a$ ,  $b$ ?

*Bảng 1. Các bước sửa chương trình “`VD_ptb1.py`”*

- 1) Bổ sung tham số  $a$ ,  $b$  vào trong cặp ngoặc () ở dòng khai báo hàm, để được `ptb1(a, b)`.
- 2) Xoá trong thân hàm hai lệnh nhập hệ số  $a$ ,  $b$  từ bàn phím.
- 3) Thay lời gọi `ptb1()` bằng `ptb1(5, 4)` để hàm thực hiện với  $a = 5$ ,  $b = 4$ .
- 4) Thêm các lời gọi thực hiện hàm `ptb1(a, b)` tương ứng với cặp hệ số  $a = 0$ ,  $b = 0$  và  $a = 0$ ,  $b = 4$ .

Một hàm có thể được thực hiện với những giá trị do chương trình truyền vào qua lời gọi hàm, tương ứng với danh sách tham số. Hai ví dụ sau đây cho thấy hai cách truyền dữ liệu cho hàm thực hiện.

- Cách thứ nhất, chương trình gọi thực hiện hàm với các giá trị cụ thể (*Ví dụ 1*).
- Cách thứ hai, chương trình gọi thực hiện hàm với giá trị tham số truyền vào (*Ví dụ 2*).

*Ví dụ 1.* Ở chương trình “`Try1_ptb1.py`”, lời gọi `ptb1(5, 4)` đã làm hàm `ptb1(a, b)` được thực hiện với  $a = 5$ ,  $b = 4$ .

*Ví dụ 2.* Chương trình ở *Hình 4* khai báo và sử dụng hàm `BMI(h, w)` tính chỉ số sức khoẻ BMI theo hai tham số chiều cao và cân nặng. Lời gọi `BMI(cao, nang)` đã làm hàm `BMI(h, w)` được thực hiện với  $h$  có giá trị của biến `cao`,  $w$  có giá trị của biến `nang`. Giá trị của hai biến `cao` và `nang` của chương trình đã được nhập vào từ bàn phím trước khi chương trình gọi thực hiện hàm `BMI(h, w)`.

```

File Edit Format Run Options Window Help

def BMI(h,w):
    idx = w/(h/100)**2
    print(' Chỉ số BMI:', idx)

cao = int(input(' Chiều cao(cm) : '))
nang = float(input(' Cân nặng(kg) : '))
BMI(cao,nang)

```

Thân hàm với các tham số để nhận giá trị từ lời gọi hàm truyền vào

Các câu lệnh nhập dữ liệu cho chương trình

Lời gọi hàm với các đối số truyền vào

Hình 4. Một ví dụ về lời gọi hàm và truyền dữ liệu cho hàm

#### 4 ► Lời gọi hàm

Trong nhiều ngôn ngữ lập trình bậc cao, hàm có thể trả về cho chương trình một giá trị qua tên của nó. Như vậy tên hàm được sử dụng như một biến trong chương trình gọi nó. Đó cũng là lí do làm cho người lập trình nhận thấy việc sử dụng hàm rất hữu ích ở nhiều trường hợp. Trong Python cũng vậy, một hàm có thể trả về một giá trị qua tên của nó nếu như có lệnh **return <Giá\_trị>** trước khi ra khỏi hàm.

Ví dụ 3. Hình 5 minh họa một ví dụ khai báo hàm có trả về giá trị và lời gọi hàm.

Tên hàm

```

File Edit Format Run Options Window Help

def chieu_cao (m, cm):
    tong = 100*m + cm
    return tong
# Nhập dữ liệu để đổi sang đơn vị centimet
so_met = int(input(' do được số met: '))
so_cm = int(input(' do được số xăng ti met: '))
# Xuất ra kết quả của hàm
print(chieu_cao(so_met, so_cm))

```

Giá trị trả về của hàm

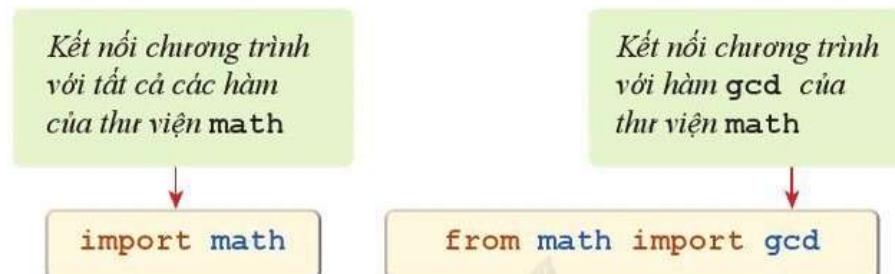
Hình 5. Một ví dụ về trả giá trị của hàm qua tên hàm

#### 5 ► Các hàm được xây dựng sẵn

Để đáp ứng tốt hơn nhu cầu đa dạng trong xử lý thông tin và giúp cho việc lập trình thuận lợi, một loạt các hàm được xây dựng sẵn, gắn với mỗi hệ thống ngôn ngữ lập trình bậc cao. Mỗi tập hợp gồm một số các hàm được xây dựng sẵn thường gọi là một thư viện. Trong chương trình của mình, người lập trình chỉ cần gọi hàm có sẵn (trong một thư viện) thực hiện mà không cần phải tự mình xây dựng lại hàm đó. Số lượng thư viện, số hàm trong mỗi thư viện, lời gọi tới chúng,... có thể thay đổi theo thời gian và phụ thuộc vào hệ thống ngôn ngữ lập trình.

Ngay từ những bước đầu tiên làm quen với lập trình Python, em đã sử dụng một số hàm trong thư viện chuẩn của Python như `print()`, `input()`,... và một số hàm toán học trong thư viện `math`. Thư viện `math` cung cấp các hằng và hàm toán học, ví dụ hàm `gcd(x, y)` trả về ước chung lớn nhất của `x` và `y`.

Để có thể sử dụng các hàm trong thư viện cần kết nối thư viện hoặc hàm đó với chương trình. Hai cách thông dụng để kết nối hàm và thư viện được nêu ở *Hình 6*.



*Hình 6. Ví dụ về hai cách kết nối với thư viện*

*Ví dụ 4.* Chương trình ở *Hình 7* kết nối hàm `gcd` trong thư viện `math`.

The screenshot shows a Python code editor with two panes. The left pane contains the following code:

```
File Edit Format Run Options Window Help
from math import gcd
a = int(input('a = '))
b = int(input('b = '))
print('Ước chung lớn nhất:', gcd(a,b))
```

The right pane shows the output of the program when run with inputs `a = 12` and `b = 20`:

```
File Edit Shell Debug Options Window Help
a = 12
b = 20
Ước chung lớn nhất: 4
>>>
```

A red bracket highlights the output line `Ước chung lớn nhất: 4`. A green box at the bottom right contains the text "Một kết quả của chương trình".

*Hình 7. Chương trình tìm ước chung lớn nhất*



**Bài 1.** Với hàm BCNN được xây dựng ở chương trình sau đây (*Hình 8*), trong những dòng lệnh có sử dụng hàm BCNN, dòng lệnh nào đúng, dòng lệnh nào sai và tại sao?

The screenshot shows a Python code editor with the following code:

```
File Edit Format Run Options Window Help
from math import gcd
def BCNN (x,y):
    return x*y//gcd(x,y)
a = int(input('a = '))
b = int(input('b = '))
print ('Bội chung nhỏ nhất:',BCNN(a,b))
c = a + b + BCNN()
print ('c = ',c)
```

*Hình 8. Chương trình về hàm bội số chung nhỏ nhất*

**Bài 2.** Chương trình ở *Hình 9* xây dựng một hàm tính diện tích một tam giác bằng công thức Heron theo ba cạnh của tam giác. Em hãy hoàn thiện chương trình bằng lời gọi hàm thích hợp để đưa ra màn hình kết quả tính diện tích của tam giác có ba cạnh là 3, 4, 5.

```
File Edit Format Run Options Window Help
def dientichtg (a, b, c):
    p = (a + b + c) /2
    s = p*(p - a)*(p - b)*(p - c)
    return s**0.5
```

*Hình 9. Chương trình tính diện tích tam giác*



Sử dụng kết quả của Bài 2 phần *Luyện tập*, em hãy viết chương trình giải bài toán ở *Hoạt động 1*.



Trong các câu sau đây, những câu nào đúng?

- 1) Sử dụng chương trình con sẽ làm chương trình dễ hiểu, dễ tìm lỗi hơn.
- 2) Hàm chỉ được gọi một lần duy nhất ở chương trình chính.
- 3) Hàm luôn trả một giá trị qua tên của hàm.
- 4) Python chỉ cho phép chương trình gọi một hàm xây dựng sẵn trong các thư viện của Python.
- 5) Khai báo hàm trong Python luôn có danh sách tham số.

### Tóm tắt bài học

- ✓ Chương trình con là một đoạn câu lệnh thực hiện một việc nào đó được đặt tên.
- ✓ Với những hàm trả về giá trị xử lý qua tên hàm thì tên của hàm được dùng như một biến trong chương trình gọi.
- ✓ Mỗi hệ thống lập trình của một ngôn ngữ lập trình bậc cao đều cung cấp một số thư viện các chương trình con được xây dựng sẵn.
- ✓ Trong Python:
  - Cách khai báo chương trình con:  
`def tên_hàm(Tham_số_1, Tham_số_2,..., Tham_số_N):`
  - Muốn xây dựng hàm trả về giá trị xử lý, cần kết thúc hàm bằng câu lệnh `return` cùng với biểu thức hay biến chứa giá trị trả về.