# GIỚI THIỆU VỀ LẬP TRÌNH

# NGÔN NGỮ LẬP TRÌNH PYTHON

## Giới thiệu về lập trình:

### a. Lập trình:

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

Những người làm nghề lập trình được gọi là các lập trình viên.

### b. Một số ứng dụng của lập trình.

Sản phẩm của nghề lập trình có ứng dụng vô cùng rộng rãi. Hầu hết các thiết bị vô tri vô giác có thể hoạt động được đều cần đến đóng góp của nghề lập trình. Tiêu biểu một số ví dụ sau:

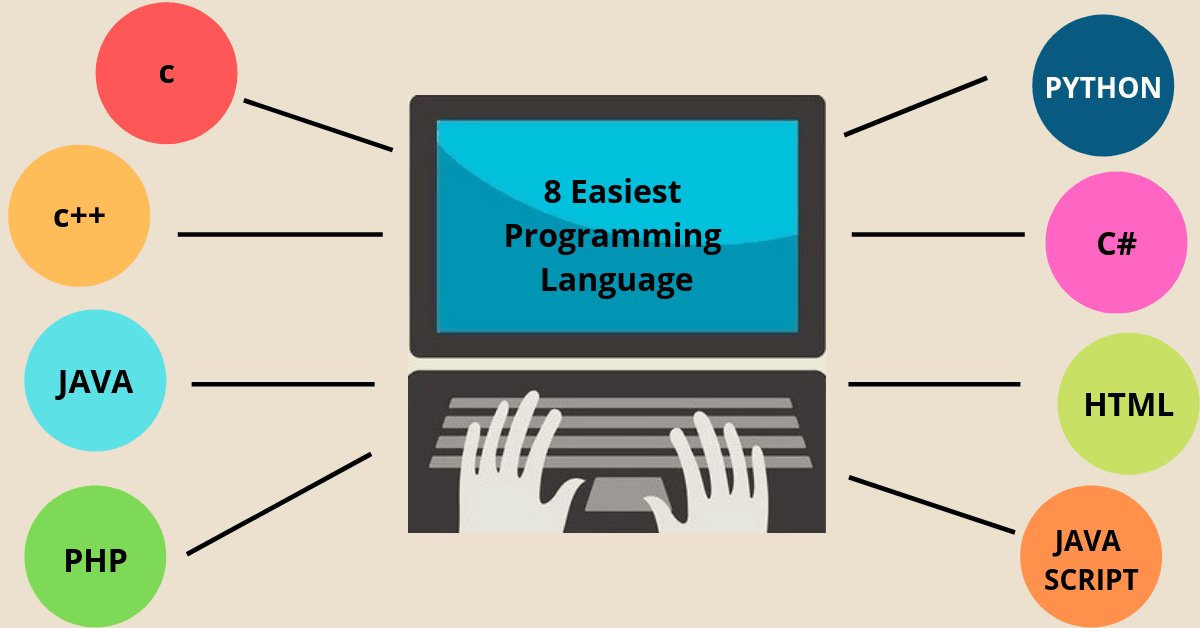
* Hệ thống website
* Các ứng dụng trả lời tự động
* Hỗ trợ dây truyền sản xuất tự động.
* Hệ thống quản lý và hỗ trợ ra quyết định.
* Hệ thống học tập trực tuyến
* Các trò chơi điện tử.

Clip: <https://youtu.be/QvyTEx1wyOY>

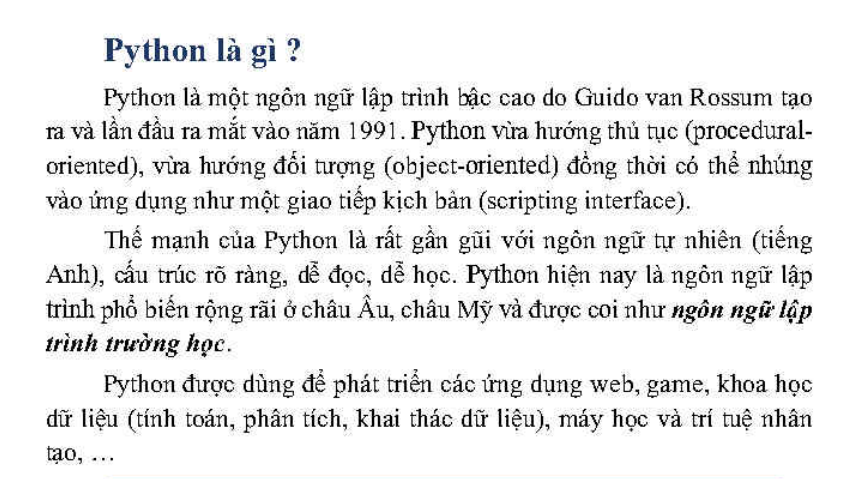
### c. Những chuyên ngành trong ngành lập trình.

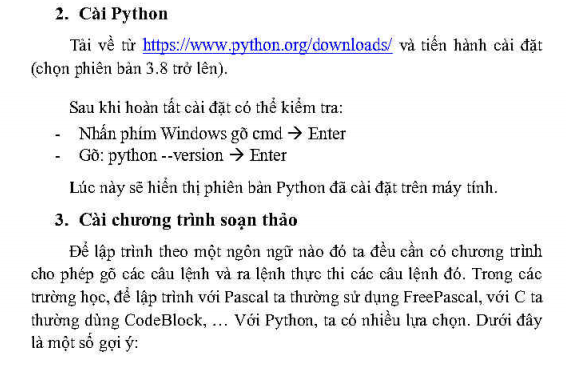
* Lập trình trên thiết bị di động
* Lập trình web
* Chuyên ngành hệ thống thông tin
* Kiểm thử phần mềm (tester)
* Chuyên gia dữ liệu.
* Machine Learning
* Lập trình IOT

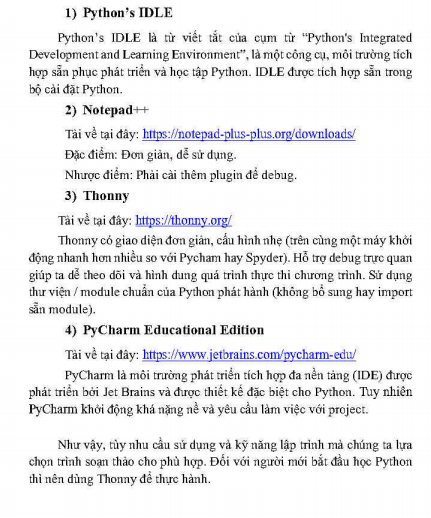
## 2. Những Ngôn ngữ lập trình phổ biến.

****

**Figure 1: Các loại NNLT bậc cao phổ biến**

****

****

****

**GDB ONLINE**

| Bài 1 | MỘT SỐ KHÁI NIỆM CƠ BẢN |
| --- | --- |

## Một số khái niệm:

### c. Chương trình:

****

### a. Thuật toán:

Thuật toán là một [tập hợp hữu hạn](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_h%E1%BB%A3p#L%E1%BB%B1c_l%C6%B0%E1%BB%A3ng_c%E1%BB%A7a_t%E1%BA%ADp_h%E1%BB%A3p_-_H%E1%BB%AFu_h%E1%BA%A1n_v%C3%A0_v%C3%B4_h%E1%BA%A1n) các **hướng dẫn** [được xác định **rõ ràng**](https://vi.wikipedia.org/w/index.php?title=%C4%90%C6%B0%E1%BB%A3c_x%C3%A1c_%C4%91%E1%BB%8Bnh_r%C3%B5&action=edit&redlink=1) **và thứ tự**, sao cho từ Input, theo các hướng dẫn ta tìm ra được Output

**\*Các đặc trưng của thuật toán:**

**- Input / Output:** mọi thuật toán dù đơn giản đến mấy cũng phải nhận dữ liệu đầu vào, xử lý nó và cho ra kết quả cuối cùng.



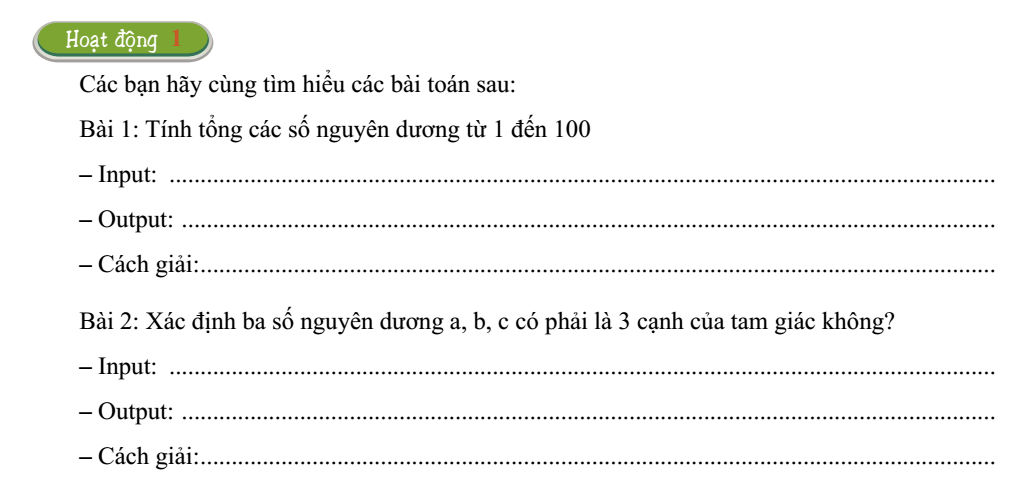
Input Output

**- Tính dừng**: Sau một số hữu hạn bước thuật toán phải dừng

**- Tính đúng đắn**: thuật toán phải đúng trong tất cả các trường hợp có thể xảy ra.

**- Tính hiệu quả**: được đánh giá dựa trên một số tiêu chuẩn như khối lượng tính toán, không gian, thời gian được sử dụng.

**Bài tập**:



### b. Ngôn ngữ lập trình:

Là ngôn ngữ dùng để trình bày thuật toán cho máy tính hiểu và thực hiện được.

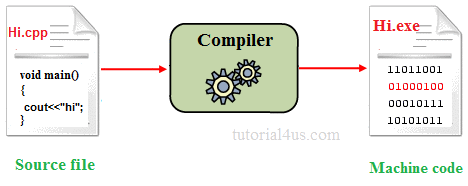
Có 2 loại ngôn ngữ lập trình cơ bản: ngôn ngữ máy, ngôn ngữ lập trình bậc cao

**Ví dụ**:

| Hãy thiết kế cách tìm vận tốc của 1 người di chuyển với vận tốc đều trên quãng đường và thời gian cho trước | Thuật toán biểu diễn băng lưu đồ | Chương trình |
| --- | --- | --- |
| **Input:** 2 số nguyên dương **s** (quãng đường, đơn vị tính: km),  **t** (thời gian, đơn vị tính: phút)  **Output:** 1 số thực **v** (vận tốc, đơn vị tinh: km/phút) |  | **Kết quả khi chạy:** |

### d. Chương trình dịch:

Có nhiệm vụ chuyển đổi chương trình từ ngôn ngữ lập trình này (CT nguồn) sang chương trình tương đương trên ngôn ngữ lập trình khác (CT đích − thường là ngôn ngữ máy)



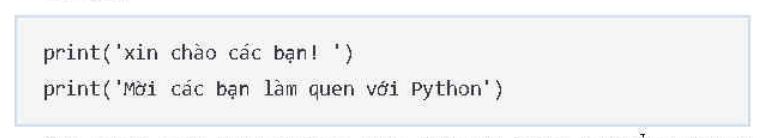
## Các thành phần cơ bản của ngôn ngữ lập trình:

***a. Bộ kí tự:*** Tập các kí tự được dùng để viết chương trình. 🡪 mã Unicode

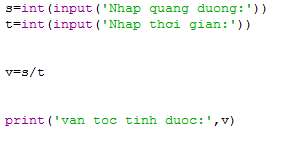
***b. Cú pháp:*** Bộ quy tắc để viết chương trình. Nhờ đó có thể mô tả chính xác thuật toán để máy thực hiện

***c. Ngữ nghĩa:*** Xác định ý nghĩa các thao tác cần thực hiện 🡪 đơn nghĩa

**ví dụ 1:**



**Ví dụ 2:**

****

# Bài 2: XUẤT DỮ LIỆU

## Nội dung bài học:

1. **Xuất kết quả ra màn hình**

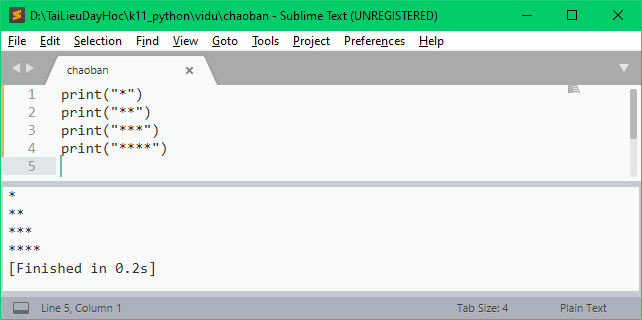
**Cú pháp:**

**print(<thông tin 1>, < thông tin 2>,…, < thông tin n>)**

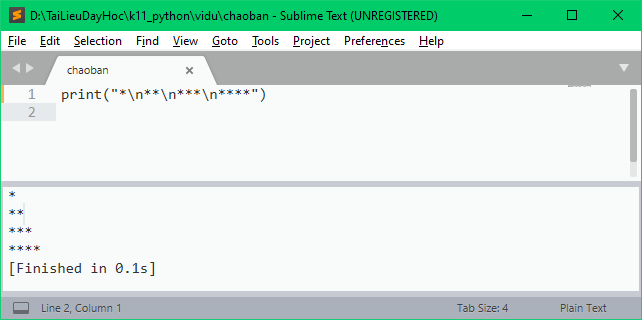
🡪**Đưa n thông tin ra màn hình**. Các **thông tin** sẽ được đưa ra trên một dòng và giữa các **thông tin** là một khoảng trắng

**Một số ví dụ:**

Ví dụ 1:

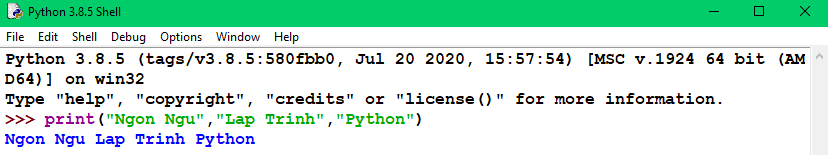


Ví dụ 2:



* Kí tự xuống dòng và về đầu dòng là ‘ **\n**’

Ví dụ 3:



* **Nhận xét**: Khi đưa nhiều **thông tin** trong câu lệnh **print(),** ta sử dụng dấu phẩy để phân cách giữa chúng.

1. **Câu lệnh print() với các tham số:**

* Tham số **sep**

**print(<thông tin 1>, < thông tin 2>,…, < thông tin n>, sep=[nội dung giữa các thông tin])**

Ví dụ:

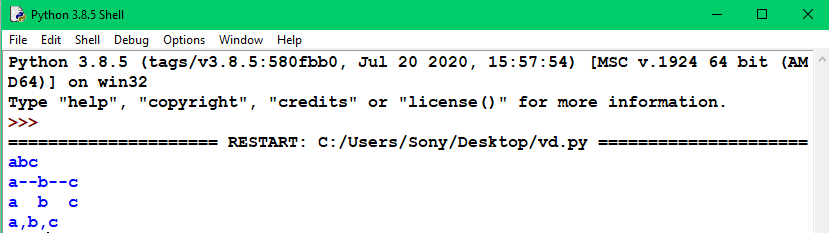
print(“a”,”b”,”c”, sep = “”)

print(“a”,”b”,”c”, sep = “--”)

print(“a”,”b”,”c”, sep = “ ”)

print(“a”,”b”,”c”, sep = “,”)

**🡺Kết quả:**

****

1. **Chú thích:**

Khi viết chương trình, để làm rõ ý nghĩa thực hiện của một sô câu lệnh nào đó, ta có thể sử dụng chú thích. Khi thực hiện chương trình, trình thông dịch sẽ bỏ qua chú thích này.

Cách viết chú thích trên một dòng: sử dụng dấu #

# nội dung cần chú thích

**Ví dụ:**

x= “Tự học Python”

# xuất biến x ra màn hình

print(x)

Cách viết chú thích trên nhiều dòng: Sử dụng nhiều lần chú thích một dòng hay cặp 3 dấu nháy đôi, hoặc cặp 3 dấu nháy đơn

‘’’……

Nội dung cần cú thích

…….

‘’’

## Vận dụng:

#### Viết 1 chương trình thực hiện theo các ý trong lưu đồ sau, chú ý: các Output xuất trên những dòng khác nhau

| **Python Program** | **Flowchart** |
| --- | --- |
|  |  |

#### Viết chương trình tính toán và xuất kết quả như hình

| **Python Program** | **Output** |
| --- | --- |
|  |  |

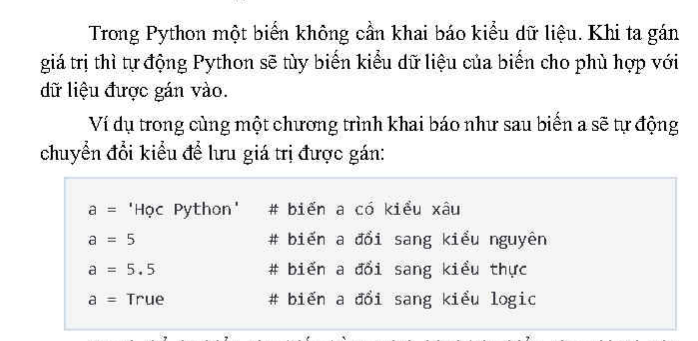
#### Viết chương trình xuất kết quả như hình

| **Python Program** | **Output** |
| --- | --- |
|  |  |

# BÀI 3 : KIỂU DỮ LIỆU TRONG PYTHON

## Nội dung bài học:

Các kiểu dữ liệu đơn giản trong Python bao gồm: chuỗi kí tự (str), số nguyên (int), số thực (float), xâu(str)

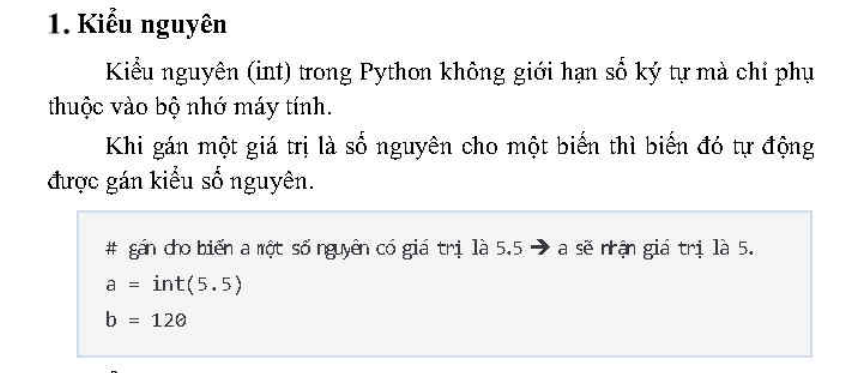


**KIỂU DỮ LIỆU SỐ**:Có 3 kiểu dữ liệu số :

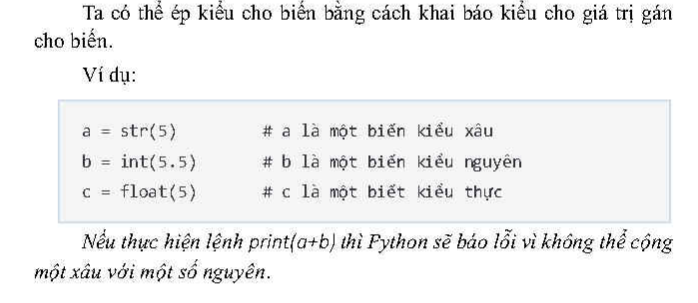
Số nguyên - int

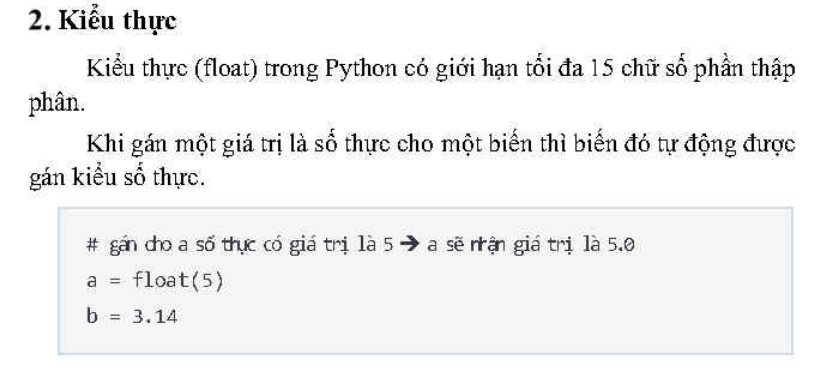
Số thực - float

Số phức – complex



Hàm int(n): ép kiểu số thực n về kiểu số nguyên





Hàm float(n): ép kiểu số nguyên n về kiểu số thực

**3. Các phép toán với kiểu số:**

**Phép toán số học:**

| Phép toán | Kí hiệu | Ví dụ |
| --- | --- | --- |
| Cộng | + |  |
| Trừ | - |  |
| Nhân | \* |  |
| Lũy thừa | \*\* | 5\*\*2 = 25 |
| Chia | / | 5/2=2.5 |
| Chia lấy thương | // | 5//2=2 |
| Chia lấy dư | % | 5%2=1 |

**Phép toán so sánh**: ==, != , < , <= , > ,>= 🡪 kết quả trả ra là kiểu LOGIC

🡪Hỗ trợ dạng so sánh kép như:

x = 2

1<x<3 # True

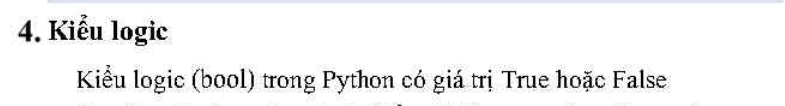
3 > x <=2 # True

2==x<4 # True

10 <x<20 # False

🡪Phép toán so sánh cho kết quả **True (đúng), False (sai)** 🡪 **kiểu LOGIC**

**Chú ý:** trong Python 3.X, kiểu dữ liệu số nguyên là vô hạn. Điều này cho phép tính toán với những số cực kì lớn, điều mà đa số các ngôn ngữ lập trình khác KHÔNG THỂ.



**Hàm type(): giúp ta biết kiểu dữ liệu của biến.**

**Cú pháp:**

**type(bien)**

**Ví dụ:**

**5. Một số hàm toán học thông dụng:**

| Tên hàm | Thực hiện |
| --- | --- |
| sqrt(x) | Tính căn bậc 2 của x |
| trunc(x) | Tính phần nguyên của x |
| pi | Tính giá trị của số pi |
| round(x,n) | Làm tròn x đến n chữ số thập phân |
| abs(x) | Tính giá trị tuyệt đối của x |
| max(a1,a2,…,an) | Trả về giá trị lớn nhất trong các số hạng a1,..,an |
| min(a1,a2,…,an) | Trả về giá trị nhỏ nhất trong các số hạng a1,..,an |

***Chú ý 1*:** để sử dụng các hàm toán học trong Python (trừ hàm abs(), min(), max(), round() ), ta cần sử dụng câu lệnh: **from math import**

ví dụ: muốn sử dụng hàm **pi** ta làm như sau:

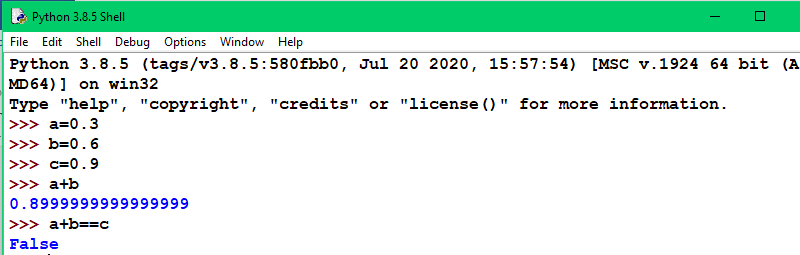
**from math import pi**

dt=3\*pi

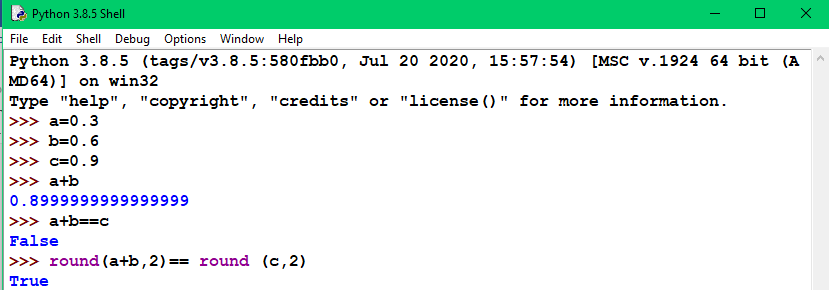
***Chú ý 2:*** để import tất cả các hàm trong thư viện math, ta thực hiện: **from math import \***

***Chú ý 3:*** Khi thực hiện tính toán với các số thực, máy tính dẫn đến những sai lệch về giá trị, ta nên sử dụng hàm làm tròn **round()** để có thể quản lý những kết quả nhận được.

Ví dụ: Nhập chương trình sau vào cửa số SHELL để thấy rõ điều này:

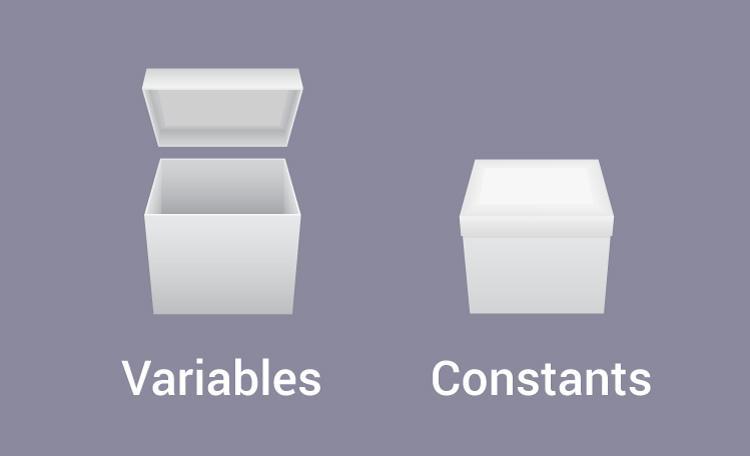
**Chương trình 1**:

**Chương trình 2:**



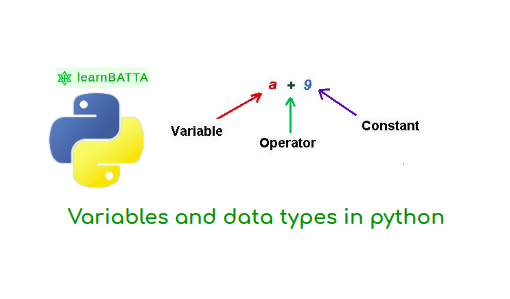


# Bài 4 BIẾN & HẰNG

****







# Tại sao lại cần biến?

Biến giúp chúng ta **lưu trữ các dữ liệu** và cho phép chúng ta **lấy các dữ liệu** của chúng để tính toán được thuận tiện và chính xác hơn.

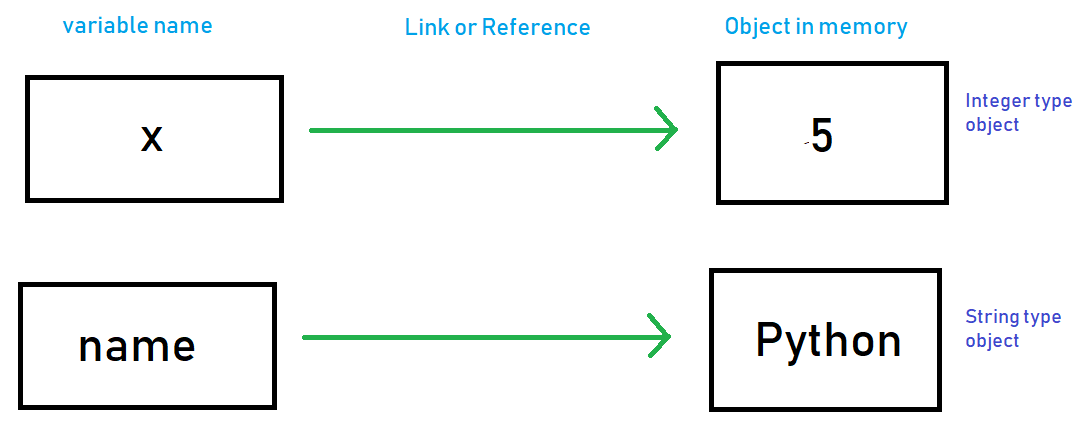
1. **BIẾN**

* Trong Python, mỗi biến là một con trỏ chỉ đến ô nhớ chứa giá trị đã được gán cho biến đó.
* Khác với ngôn ngữ lập trình khác, Python không có lệnh khai báo biến.

**Ví dụ 1:**

x=5

name=”Python”



**Ví dụ 2:**

x=7

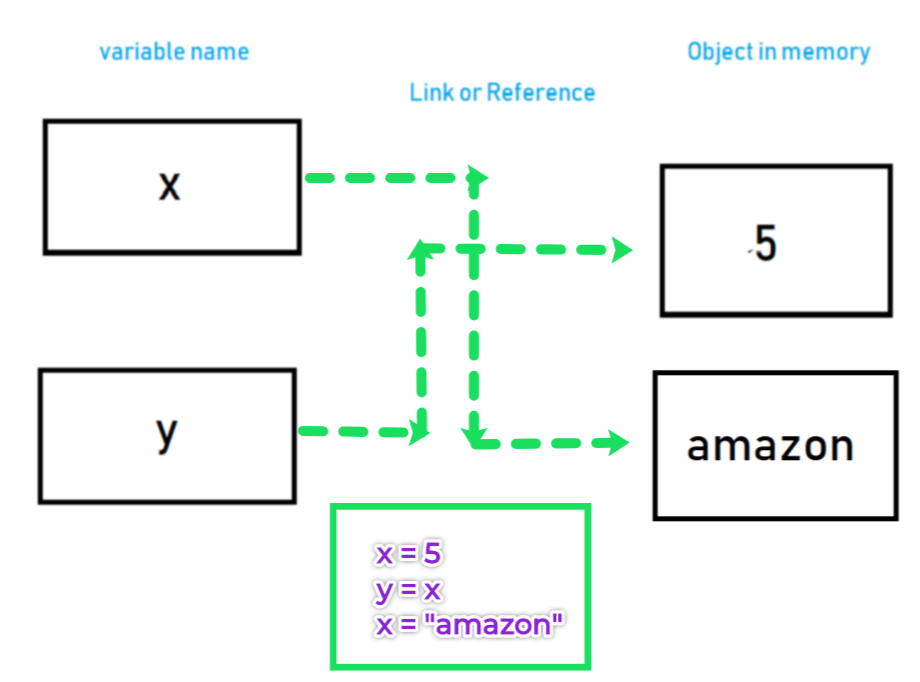
y=3.14159

z=True

t=z

| **variable name** | **object in memory** |
| --- | --- |
| **E:\NTBH\NHH\GiaoAn\Python\TongHop_Python\HinhAnh\tải xuống.png** | |

**Ví dụ 3:**

****

1. **Một số qui tắc khi đặt tên biến:**

* Biến chỉ chứa các kí tự **chữ cái, chữ số và dấu gạch dưới** \_
* **Tên biến có phân biệt chữ hoa, thường.** Ví dụ: PI, Pi, pI, pi là 4 tên biến khác nhau.
* **Không chứa kí tự trống (space)** và các kí tự đặc biệt khác như **+,-,\*….**
* Tên biến **không bắt đầu bằng kí tự số**
* Không sử dụng các từ khóa, tên hàm đã được ngôn ngữ lập trình sử dụng
* Tên biến nên đặt ngắn gọn, gợi nhớ, dễ nhớ

**Ví dụ:**

Tên đúng: x, y, x1, x2

Tên sai: 1x, a b, a?b

1. **GÁN GIÁ TRỊ CHO BIẾN**

* **Mục đích:** Dùng để thay đổi giá trị hiện thời của biến. Hoặc dùng khi muốn lưu trữ giá trị của một biểu thức tính toán.
* **Cú pháp:**

<Tên biến> = <Giá trị>

Ví dụ:

x = 3

x = “python”

**\* Lưu ý với phép gán:**

+ **Bên trái bắt buộc là biến, bên phải là biểu thức hoặc giá trị**

* Các phép gán sai:

x+3=y

5=y

5\*\*3=8+5

5==6 (đây là phép so sánh, không phải phép gán)

1. **HẰNG**:

Là một đại lượng **có giá trị không đổi** trong suốt chương trình. Có 3 loại hằng: **hằng số** (5, 7,….), **hằng logic** (True, False)**,** **hằng chuỗi** (‘hello’, “11A1”)

1. **LUYỆN TẬP**
2. **Xét tính đúng sai của các chương trình sau. Nếu sai hãy giải thích:**

Chương trình 1:

a = b

Chương trình 2:

a = 1

a = b

Chương trình 3:

a = 1

b = a

Chương trình 4:

a = 2020

2020 = a

1. **Không chạy chương trình, hãy đưa ra kết quả của chương trình sau:**

Chương trình 1:

x=2020

print(x)

print(“x”)

Chương trình 2:

x= 2020

print (x+X)

# **BÀI 5** : NHẬP DỮ LIỆU TỪ BÀN PHÍM

## DẪN NHẬP

| **Python Program** |
| --- |
|  |
| **Flowchart Program** |
|  |

Thử gõ đoạn code trên và cho biết kết quả xuất ra.

## NỘI DUNG BÀI HỌC:

1. **Nhập dữ liệu từ bàn phím**

**Cú pháp:**

<Tên biến>=**input**(‘Chuoi thông báo nhập’)

* Dùng để lưu một xâu kí tự được nhập từ bàn phím vào <Tên biến>

**Một số ví dụ:**

**Ví dụ 1:**

| x= input(“bạn tên gì”)  print(“tôi tên”,x) |  |  |
| --- | --- | --- |

Ví dụ 2: Nhập 2 số nguyên a và b. Đưa ra tổng a +b

| a= int(input('nhập giá trị a:'))  b= int(input('nhập giá trị b:'))  print(' a+b=',a+b) |  |  |
| --- | --- | --- |



Chú ý: Câu

**BÀI TẬP & THỰC HÀNH**

| **Chuẩn bị**   * **B1: Xác định Input, Output** * **B2: Xác định thuật toán, công thức**   **Thực hiện**   * Nhập Input * Tính toán theo thuật toán, công thức đã xác định * Xuất Output |
| --- |

#### Viết chương trình nhập vào 3 số nguyên a,b,c. Cho biết tổng, tích, tbc của 3 số đó.

#### Viết chương trình nhập vào 2 số nguyên a,b. Hoán đổi giá trị 2 biến đó

#### Viết chương trình xác định kết quả biểu thức: x2-4 **>=**y2 (với x,y là 2 số thực nhập từ bàn phím)

#### Viết chương trình nhập vào bán kính hình tròn, tính chu vi, diện tích hình tròn đó. In ra màn hình kết quả tính được lấy 1 số lẻ thập phân.

#### Viết chương trình nhập vào giờ, phút, giây ( hh : mm : ss ) từ bàn phím . In ra tổng số giây.

#### Viết chương trình nhập vào 1 số nguyên dương có 2 chữ số. In ra số đảo ngược của số đó.

#### Viết chương trình tích của 2 phân số. Với tử số, mẫu số được nhập từ bàn phím.

#### Viết chương trình: nhập vào ngày, tháng, năm sinh của 1 nhân viên nam. In ra ngày về hưu của nhân viên đó. Biết tuổi về hưu đối với nam là 60.

#### Viết chương trình: Nhập vào 1 số tiền nguyên N đồng. Đổi ra xem được bao nhiêu tờ 10đ, 5đ, 2đ, 1đ

#### Ví dụ: N= 543 đ 🡺 54 tờ 10đ, 0 tờ 5đ, 1 tờ 2đ, 1 tờ 1đ

#### Viết chương trình đổi các đơn vị sau:

### **kg->pound (1kg = 2.20462262 pounds),**

### **m🡪feet (**1 meter = 3.2808399 feet),

### km🡪miles (1 km = 0.6214 miles )

#### Nhập vào 1 số nguyên dương có 4 chữ số. Làm các công việc sau. Xuất ra màn hình chuỗi của việc đọc giờ số này:

#### VD: 3670: 1giờ 1 phút 10 giây

#### Nhập vào 1 số nguyên dương có 3 chữ số. Làm các công việc sau:

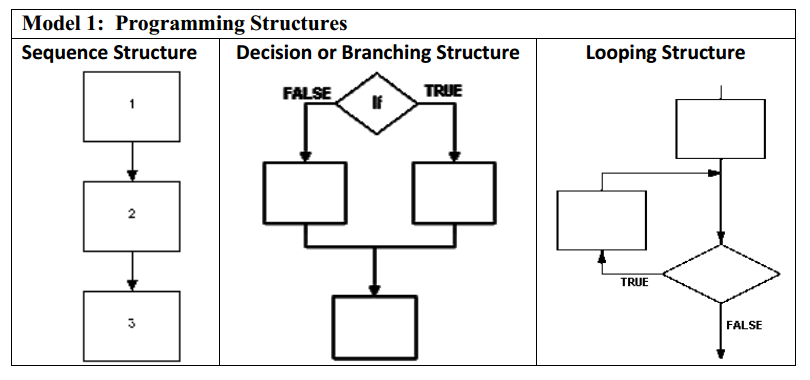
### + Xuất ra màn hình chuỗi của việc đọc số này: VD: 358: 3 trăm 5 chục 8 đơn vị

### +Tính và in ra màn hình tổng các chữ số đó VD: 358= 3+5+8=16

#### Viết chương trình Tính tiền điện tiêu thụ biết giá 750đ/kwh với chỉ số cũ và mới của điện kế được nhập vào từ bàn phím.

# Chương 2 CẤU TRÚC ĐIỀU KHIỂN

Trong một chương trình, bình thường các lệnh sẽ lần lượt được thực hiện theo thứ tự xuất hiện của nó trong file code. Nếu chỉ thực thi lệnh như vậy các chương trình sẽ rất hạn chế. Vì vậy người ta đưa vào các cấu trúc điều khiển có tác dụng làm **thay đổi trật tự thực thi lệnh trong chương trình**.



# BÀI 6: CẤU TRÚC RẼ NHÁNH

Cấu trúc rẽ nhánh **quyết định** xem những **lệnh nào sẽ được thực hiện** **căn cứ vào** giá trị (kiểu bool) của biểu thức **điều kiện** **cho trước**

## Câu lệnh **if**

**Cú pháp:**

| **if** <biểu thức điều kiện> :  <Khối lệnh>  **ý nghĩa thực hiện**  Nếu <biểu thức điều kiện> đúng thì sẽ thực hiện <Khối lệnh> |  |
| --- | --- |

Trong đó:

* <biểu thức điều kiện>: là biểu thức có sử dụng các phép toán so sánh (==,!=,>,>=,<,<=), các phép toán logic (NOT, AND, OR). **Kết quả của biểu thức điều kiện là True Hoặc False**

🡪Hỗ trợ dạng so sánh kép như:

x = 2 #Câu lệnh gán

1<x<3 # True

3 > x <=2 # True

2==x<4 # True

10 <x<20 # False

* <Khối lệnh>: là gồm **một** lệnh **hoặc nhiều lệnh**  được **viết thẳng cột** với nhau *(mỗi lệnh trên 1 dòng)*. Các lệnh này được gọi là cùng mức (level)

**Ví dụ 1:** chương trình:

a= int (input())

b= int (input())

if a<b :

print(“a nhỏ hơn b”)

**Ví dụ 2:** chương trình:

a= int (input())

b= int (input())

if a<b :

print(“a nhỏ hơn b”)

print(“ a + b = ”, a+b)

**Ví dụ 3:** chương trình

a= int (input())

b= int (input())

if a > b :

print(“a lớn hơn b”)

print(“ a - b = ”, a - b)

print (“ a + b = ”, a + b) # câu lệnh này ngoài If

## Câu lệnh **if… else**…

**Cú pháp:**

| **if** <biểu thức điều kiện> :  <Khối lệnh 1>  **else:**  <Khối lệnh 2>  **Ý nghĩa thực hiện:**   * Nếu <biểu thức điều kiện> **ĐÚNG** thì sẽ thực hiện <Khối lệnh 1>, bỏ qua <khối lệnh 2> * Nếu <biểu thức điều kiện> **SAI** thì sẽ thực hiện <Khối lệnh 2> bỏ qua <khối lệnh 1> |  |
| --- | --- |

**Ví dụ:**

a= int (input())

b= int (input())

if a == b :

print(“a và b bằng nhau”)

else:

print(“a và b khác nhau”)

## Câu lệnh **if** lồng nhau:

**Cú pháp:**

| **if** <biểu thức điều kiện 1> :  <Khối lệnh 1>  **else:**  **if <biểu thức điều kiện 2>:**  <Khối lệnh 2>  else:  <Khối lệnh 3> | **if** <biểu thức điều kiện 1> :  <Khối lệnh 1>  **elif <biểu thức điều kiện 2>:**  <Khối lệnh 2>  else:  <Khối lệnh 3> |  |
| --- | --- | --- |

**Thực hiện:**

* Nếu <biểu thức điều kiện 1> **ĐÚNG** thì sẽ thực hiện <Khối lệnh 1> bỏ qua **elif**
* Nếu <biểu thức điều kiện 1> **SAI** thì bỏ qua <khối lệnh 1>; kiểm tra <biểu thức điều kiện 2>
  + **ĐÚNG** sẽ thực hiện <Khối lệnh 2>
  + **SAI** thực hiện <Khối lệnh 3>

**Ví dụ 1:**

a= int (input())

b= int (input())

**if** a == b :

print(“a và b bằng nhau”)

**else:**

**if** a < b:

print(“a nhỏ hơn b”)

else:

print(“a lớn hơn b”)

## Bài tập:

* 1. Viết chương trình nhập vào 1 số nguyên. Kiểm tra xem số đó chẵn hay lẻ. In ra màn hình thông báo tương ứng
  2. Viết chương trình giải pt bậc 1: ax+b=0 (với a,b là 2 số nguyên được nhập từ bàn phím)
  3. Viết chương trình giải pt bậc 2: ax2+bx+c=0 (với a,b,c là các số nguyên được nhập từ bàn phím)
  4. Viết chương trình nhập hai số thực . Sau đó hỏi phép tính cần thực hiện và in kết quả của phép tính đó .

Nếu là “+” , in kết quả của tổng lên màn hình .

Nếu là “-” , in kết quả của hiệu lên màn hình .

Nếu là “/” , in kết quả của thương lên màn hình .

Nếu là “\*” , in kết quả của tích lên màn hình .

* 1. Nhập 3 số a , b , c bất kì . Hãy kiểm tra xem ba số đó có thể là độ dài ba cạnh của một tam giác hay không ? Thông báo lên màn hình ‘ Thỏa ‘, ‘ Không thỏa’ trong từng trường hợp tương ứng. Nếu Là Tam giác thì kiểm tra đó là tam giác đều, tam giác cân, tam giác vuông hay không?
  2. Viết chương trình nhập vào 1 năm bất kỳ. Cho biết đó có phải là năm nhuận không(biết rằng năm nhuận hoặc ***phải*** ***chia hết cho 4 và không chia hết cho 100*** hoặc  ***phải chia hết cho 400***)
  3. Viết chương trình nhập vào 1 tháng và năm bất kỳ. Cho biết số ngày trong tháng ấy (xét năm nhuận)
  4. Nhập điểm cho 4 môn học. Tính TBC và xếp loại như sau:

8.0 <= TBC : Gioi

6.5 <= TBC < 8.0 : Kha

5.0 <= TBC < 6.5 : TB

TBC < 5.0 : Kem

* 1. Viết chương trình tính tiền điện theo Kw nhập vào như sau:

Nếu số Kw <=300 🡪 đơn giá: 500 đồng/Kw

Nếu số 300<Kw <=600 🡪 đơn giá: 700 đồng/Kw

Nếu số Kw > 600 🡪 đơn giá: 1000 đồng/Kw

**Thành tiền:** Cộng thêm 10% thuế thu nhập vào số tiền tính được

* 1. \*Viết chương trình nhập vào 1 ngày, tháng và năm bất kỳ. Cho biết ngày trước đó.
  2. \*Viết chương trình nhập vào 1 tháng và năm bất kỳ Cho biết ngày sau đó