**BÀI 3: TÌM HIỂU VỀ PYTHON CƠ BẢN**

## Mục tiêu

* Tìm hiểu về Python cơ bản

## Nội dung

### Tìm hiểu Python cơ bản

* + 1. **Một số từ khóa trong Python**

Bảng dưới liệt kê các từ khóa trong Python. Đây là các từ dành riêng và bạn không thể sử dụng chúng như là các hằng, biến hoặc cho bất kỳ tên định danh nào. Tất cả từ khóa trong Python là chỉ ở dạng chữ thường

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **and** | **def** | **exec** | **if** | **not** | **return** |
| **assert** | **del** | **finally** | **import** | **or** | **try** |
| **break** | **elif** | **for** | **in** | **pass** | **while** |
| **class** | **else** | **from** | **is** | **print** | **with** |
| **continue** | **except** | **global** | **lambda** | **raise** | **yield** |

* + 1. **Dòng lệnh và độ thụt dòng**

Python không cung cấp các dấu ngoặc bao hàm ({}) để chỉ các khối code cho định nghĩa lớp hoặc hàm hoặc điều khiển luồng. Các khối code được nhận biết bởi độ thụt dòng code (indentation) trong Python và đây là điều bắt buộc.

Số khoảng trống trong độ thụt dòng là biến đổi, nhưng tất cả các lệnh bên trong khối phải được thụt cùng một số lượng khoảng trống như nhau. Trong Python thì tất cả các dòng liên tiếp nhau mà được thụt đầu dòng với cùng lượng khoảng trống như nhau sẽ tạo nên một khối.

Ví dụ:

|  |
| --- |
| # Đây là code đúng  if True:  print “True”  print “True”  else:  print “False”  print “False”  # Đây là code sai  if True:  print “True”  print “True”  else:  print “False”  print “False”  # Đây cũng là code sai  if True:  print “True”  print “True”  else:  print “False”  print “False” |

* + 1. **Lệnh trên nhiều dòng**

Các lệnh trong Python có một nét đặc trưng là kết thúc với một newline (dòng mới). Tuy nhiên, Python cho phép sử dụng ký tự \ để chỉ rõ sự liên tục dòng. Ví dụ:

|  |
| --- |
| Sum = a +b /  + c + d /  + e |

Các lệnh được chứa bên trong các dấu ngoặc [], {}, hoặc () thì không cần sử dụng ký tự \. Ví dụ:

|  |
| --- |
| Thu = [ 2 , 3 , 4 ,  5 , 6 , 7 ] |

* + 1. **Các kiểu biến trong Python**

Biến là các vị trị bộ nhớ được dành riêng để lưu trữ dữ liệu. Một khi một biến đã được lưu trữ, nghĩa là một khoảng không gian đã được cấp phát trong bộ nhớ đó.

Dựa trên kiểu dữ liệu của một biến, trình thông dịch cấp phát bộ nhớ và quyết định những gì có thể được lưu trữ trong khu nhớ dành riêng đó. Vì thế, bằng việc gán các kiểu dữ liệu khác nhau cho các biến, bạn có thể lưu trữ số nguyên, thập phân hoặc ký tự trong các biến này.

Ví dụ:

|  |
| --- |
| A = 5 # Gán số nguyên  A = 5.0 # Gán số thực  A = “Hello Python” # Gán chuỗi |

* **Các kiểu dữ liệu trong Python**

Trong Python có 5 kiểu dữ liệu chính là : Number, String, List, Tuple, Dictionary

* + 1. **Nhập xuất trong Python**

Để xuất dữ liệu ra màn hình ta sử dụng **print**

Ví dụ :

|  |
| --- |
| **print** “Vi du su dung print” |

Output :

|  |
| --- |
| Vi du su dung print |

Ví dụ :

|  |
| --- |
| a = 5  **print** a  **print** “a = %d” %a |

Output :

|  |
| --- |
| 5  a = 5 |

Mặc định khi print xong sẽ xuống dòng. Nếu ta dung dấu phẩy sau print thì nó sẽ in ra khoảng trắng thay vì dấu xuống dòng.

Ví dụ :

|  |
| --- |
| a = 5  b = 6  **print** a  **print** b  **print** a,  **print** b |

Output :

|  |
| --- |
| 5  6  5 6 |

**Nhập dữ liệu**

Để nhập dữ liệu, chúng ta có 2 cách là input() và raw\_input()

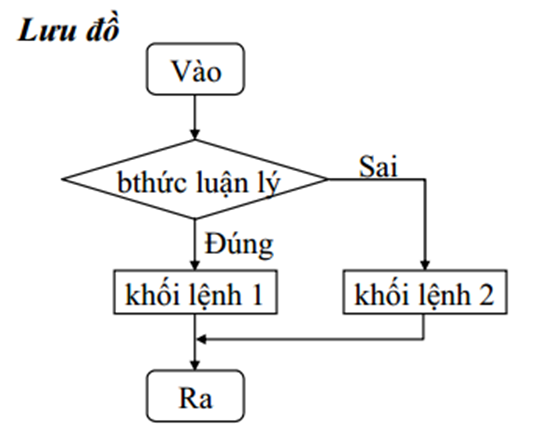
**input()** : Dữ liệu đưa vào sẽ tự định dạng kiểu

**raw\_input()** : Dữ liệu đưa vào sẽ được định dạng kiểu nguyên bản

Ví dụ :

|  |
| --- |
| ten = raw\_input(“Moi ban nhap vao ten :”)  tuoi = input(“Moi ban nhap vao tuoi : ) |

* + 1. **Cấu trúc rẽ nhánh**



Cú pháp:

|  |
| --- |
| If điều\_kiện:  Câu\_lệnh  Else:  Câu\_lệnh |

* + 1. **Cấu trúc lặp**

Khi muốn thực hiện lại một công việc nhiều lần ta sẽ sử dụng vòng lặp. Python cung cấp 2 kiểu lặp là kiểu for và while

Vòng lặp while: lặp đi lặp lại một lệnh hoặc một nhóm lệnh trong khi một điều kiện đã cho là TRUE. Nó kiểm tra điều kiện trước khi thực thi phần thân của vòng lặp.

Vòng lặp for: nó có khả năng lặp qua các item của bất kỳ dãy nào như một list hoặc một string.

Ví dụ:

|  |
| --- |
| for i in range(0,5) :  print i |

Hàm range(start, stop, step) Cho phép tạo ra 1 mảng có phần tử đầu tiên = start, bước nhảy là step và giới hạn là stop-1. Nếu không có step thì sẽ mặc định step = 1

|  |
| --- |
| >>> range(0,5)  [0,1,2,3,4] |

## Câu hỏi và bài tập

* + 1. **Viết các Function cộng, trừ, nhân, chia 2 số, sắp xếp mảng 1 chiều, mảng 2 chiều**
    2. **Viết và chạy lại chương trình sau, giải thích chức năng của từng câu lệnh**

