



LOGO

NHẬP MÔN LẬP TRÌNH CHO KHOA HỌC DỮ LIỆU

Bài 2. Lập trình căn bản với Python

Nội dung

1

Giới thiệu về ngôn ngữ Python

2

Cách thực hiện câu lệnh, chương trình

3

Biến, kiểu dữ liệu và câu lệnh

4

Nhập/xuất dữ liệu

Giới thiệu về ngôn ngữ Python

- Python lần đầu được giới thiệu vào tháng 12/1989
- Tác giả là Guido van Rossum (Hà Lan)
 - Sinh năm 1956
 - Hiện đang làm cho Google
- Python kế thừa từ ngôn ngữ ABC
- Python 2 được giới thiệu năm 2000
 - Hỗ trợ unicode
 - Mã python 2 rất phổ biến
- Python 3 được phát hành năm 2008
 - Hiện đã có phiên bản 3.7



Giới thiệu về ngôn ngữ Python

- Được xếp vào loại “ngôn ngữ kịch bản” (scripting programming language)
 - Thích hợp với DevOps (viết code cũng là vận hành)
 - Khai báo biến tự nhiên, phong phú và động
 - Nhiều phép tính cấp cao được cung cấp sẵn
 - Thường được thông dịch thay vì biên dịch
- Những người cuồng python (pythonista) cho rằng ngôn ngữ này trong sáng và tiện dụng đến mức ta có thể dùng nó cho mọi khâu lập trình (chứ không phải chỉ viết script)

Giới thiệu về ngôn ngữ Python

- Là ngôn ngữ mã nguồn mở
- Vừa hướng thủ tục, vừa hướng đối tượng
- Hỗ trợ module và hỗ trợ gói (package)
- Xử lý lỗi bằng ngoại lệ (exception)
- Kiểu dữ liệu động ở mức cao
- Có khả năng tương tác với các module viết bằng ngôn ngữ lập trình khác
- Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface)

Giới thiệu về ngôn ngữ Python

Ưu điểm

- Có ngữ pháp đơn giản, dễ đọc
- Viết mã ngắn gọn hơn những chương trình tương đương được viết trong C, C++, C#, Java,...
- Có các bộ thư viện chuẩn và các module ngoài, đáp ứng gần như mọi nhu cầu lập trình
- Có khả năng chạy trên nhiều nền tảng (Windows, Linux, Unix, OS/2, Mac, Amiga, máy ảo .NET, máy ảo Java, Nokia Series 60,...)
- Có cộng đồng lập trình rất lớn, hệ thống thư viện chuẩn, mã nguồn chia sẻ nhiều

Giới thiệu về ngôn ngữ Python

Nhược điểm

- Chương trình chạy chậm
- Giao tiếp với các thư viện viết bằng các ngôn ngữ khác tương đối khó khăn
- Yếu trong hỗ trợ tính toán trên di động
- Gỡ lỗi đòi hỏi kinh nghiệm
- Kém hỗ trợ các cơ sở dữ liệu

Cách thực hiện câu lệnh

Cài đặt

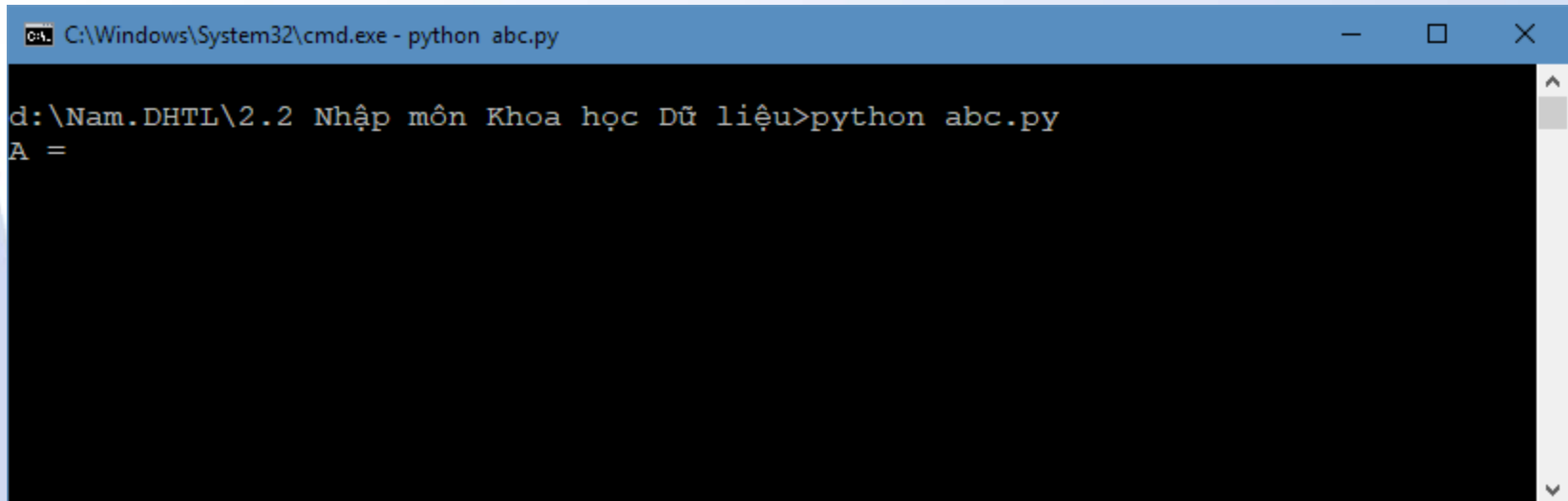


<https://www.python.org/downloads/>

Cách thực hiện câu lệnh

Khởi chạy

- Python có 2 chế độ thực thi
 - Chế độ thực thi: chỉ ra chương trình cần thực hiện
 - Trình dịch python sẽ nạp, dịch và chạy chương trình đó
 - Chế độ dòng lệnh: chạy từng lệnh một
- Chế độ thực thi: “`python abc.py`” chạy file `abc.py`



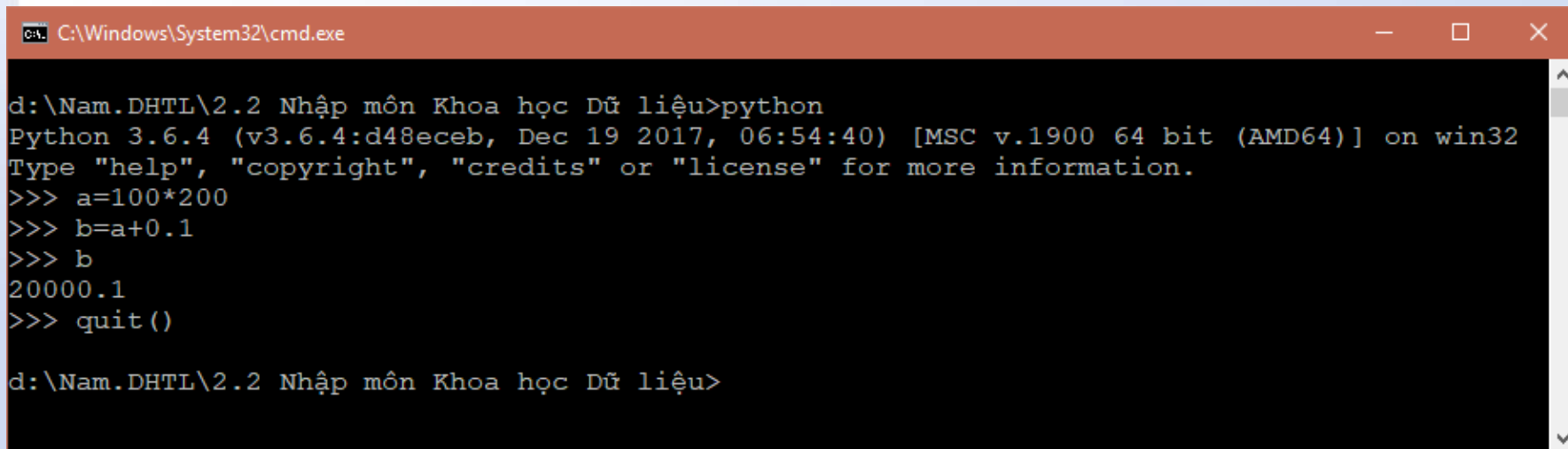
```
C:\Windows\System32\cmd.exe - python abc.py

d:\Nam.DHTL\2.2 Nhập môn Khoa học Dữ liệu>python abc.py
A =
```

Cách thực hiện câu lệnh

Chế độ dòng lệnh

- Lúc này trình thông dịch python sẽ chờ người dùng gõ từng dòng lệnh
- Gõ dòng lệnh nào xong, python chạy liền dòng đó
- Chấm dứt chế độ này bằng cách gõ lệnh: “`quit()`”



```
C:\Windows\System32\cmd.exe

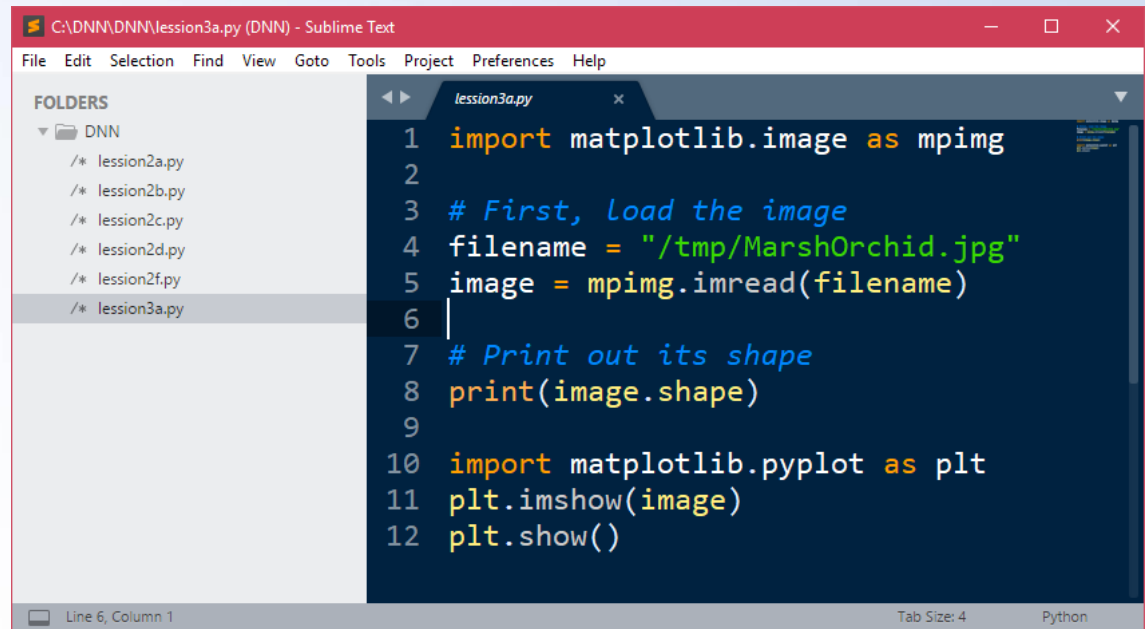
d:\Nam.DHTL\2.2 Nhập môn Khoa học Dữ liệu>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a=100*200
>>> b=a+0.1
>>> b
20000.1
>>> quit()

d:\Nam.DHTL\2.2 Nhập môn Khoa học Dữ liệu>
```

Cách thực hiện câu lệnh

Chế độ thực thi (soạn thảo)

- Làm thế nào để viết chương trình python (.py)?
 - Dùng phần mềm soạn thảo văn bản thô (txt) bất kỳ để soạn và lưu file ở dạng .py rồi dịch bằng python
- Có những phần mềm thích hợp cho việc này hơn
 - IDLE
 - Sublime Text
 - Notepad++
 - PyCharm
 - Spyder
 - Rodeo
 - ...



The screenshot shows the Sublime Text editor window titled "C:\DNN\DNN\lesson3a.py (DNN) - Sublime Text". The left sidebar displays a "FOLDERS" panel with a tree view of the "DNN" directory containing files: lesson2a.py, lesson2b.py, lesson2c.py, lesson2d.py, lesson2f.py, and lesson3a.py. The main editor area shows the content of lesson3a.py with the following code:

```
1 import matplotlib.image as mpimg
2
3 # First, load the image
4 filename = "/tmp/MarshOrchid.jpg"
5 image = mpimg.imread(filename)
6
7 # Print out its shape
8 print(image.shape)
9
10 import matplotlib.pyplot as plt
11 plt.imshow(image)
12 plt.show()
```

The status bar at the bottom indicates "Line 6, Column 1", "Tab Size: 4", and "Python".

Cách thực hiện câu lệnh

Chế độ thực thi (biên dịch mã python)

- Trường hợp cần thiết, mã python có thể được biên dịch, kết quả dịch là chương trình dạng bytecode cho máy ảo python
 - Tương tự như trường hợp của ngôn ngữ java
- Mã lệnh dịch được lưu vào file với đuôi `.pyc`
- Việc biên dịch có nhiều lợi điểm, chẳng hạn như khi sử dụng câu lệnh `import` một thư viện nào đó, thì có thể sử dụng luôn mã pyc có sẵn thay vì phải dịch lại từ đầu

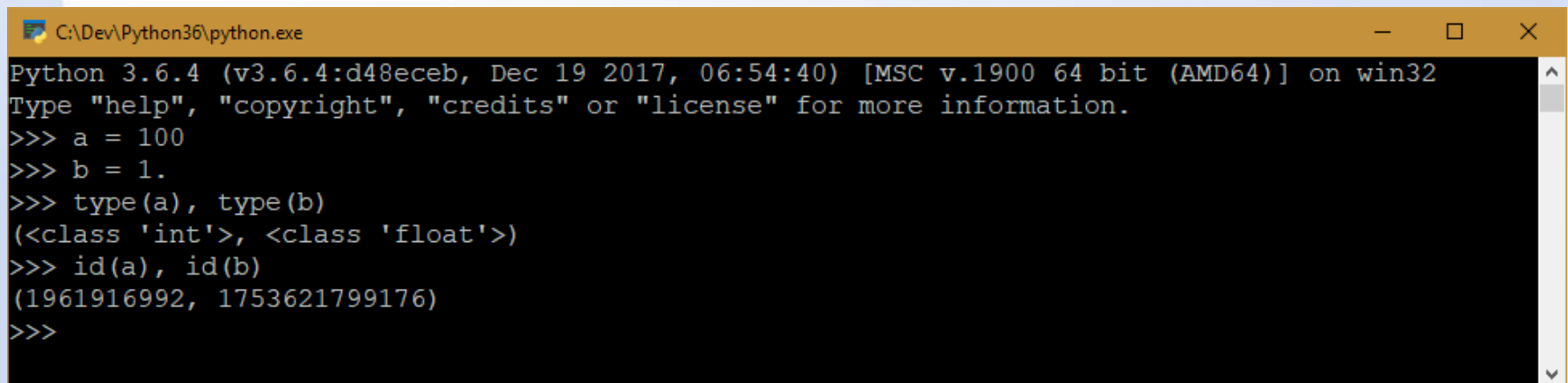
Biến

- **Biến = vùng bộ nhớ được đặt tên (để dễ thao tác)**
- **Biến trong python:**
 - Có tên, phân biệt chữ hoa/thường
 - Không cần khai báo trước
 - Không cần chỉ ra kiểu dữ liệu
 - Có thể thay đổi sang kiểu dữ liệu khác
 - Nên gán giá trị ngay khi bắt đầu xuất hiện
- **Ví dụ:**

```
n = 12          # biến n là kiểu nguyên
n = n + 0.1     # biến n chuyển sang kiểu thực
```

Biến

- Tên biến có thể chứa chữ cái hoặc chữ số hoặc gạch dưới (_), kí tự bắt đầu không được dùng chữ số
 - Không được trùng với từ khóa (tất nhiên)
 - Từ python 3 được dùng chữ cái unicode
- Tất cả mọi biến trong python đều là các đối tượng, vì thế nó có kiểu và vị trí trong bộ nhớ (id)



```
C:\Dev\Python36\python.exe
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 100
>>> b = 1.
>>> type(a), type(b)
(<class 'int'>, <class 'float'>)
>>> id(a), id(b)
(1961916992, 1753621799176)
>>>
```


Kiểu dữ liệu

Kiểu int (số nguyên)

- Kiểu Int biểu diễn số nguyên (Integer)
 - Giá trị: ..., -3, -2, -1, 0, 1, 2, 3, 4, 5, ...
 - Số nguyên có định dạng: 1, 45, 43028030 (không có dấu phẩy và dấu chấm)
 - Các toán tử: +, -, *, //, **, toán tử 1 ngôi –
- Nguyên tắc: các toán tử trên các giá trị Int phải mang kiểu Int
 - Ví dụ: 1 // 2, kết quả trả về phần nguyên 0;
 - Toán hạng thường sử dụng cùng: % (lấy phần dư)
 - Toán hạng / ko phải là phép chia trong Python3
- Trong Python 3.X, **kiểu dữ liệu số nguyên là vô hạn**, cho phép tính toán với những số cực kì lớn, điều mà đa số các ngôn ngữ lập trình khác KHÔNG THỂ.

Kiểu dữ liệu

Kiểu float (số thực)

- Kiểu float biểu diễn số thực: là tập hợp các số nguyên và số thập phân 1, 1.4, -123, 69.96,...
- Giá trị: phân biệt với số nguyên bởi các số thập phân
 - Trong Python, một số có '.' là một số thực (Ví dụ: 2.0)
- Các toán tử: +, -, *, /, **, toán tử 1 ngôi –
 - Chú ý, số thực có phép chia / khác với số nguyên //
 - Ví dụ: 1.0 // 2.0 được tính bằng 0
- Kí hiệu mũ có ý nghĩa cho các số giá trị lớn
 - 22.51e-6 là $22.51 * 10^{-6}$ hay 0.00002251
- Số thực trong Python có **độ chính xác xấp xỉ 15 chữ số phần thập phân**.
- Sử dụng decimal để có độ chính xác cao hơn **from decimal import ***

Kiểu dữ liệu

Kiểu phân số

- Phân số gồm hai phần là tử số và mẫu số.
- Tạo phân số trong Python: `Fraction(<Tử_số>, <Mẫu_số>)`
- Ví dụ: Nhập phân số $\frac{1}{4}$, $\frac{3}{9}$, $\frac{3}{4}$,

```
0 >>> from fractions import * # lấy toàn bộ nội dung của thư viện decimal
1 >>> Fraction(1, 4) # phân số với tử số là 1, mẫu số là 4.
2 Fraction(1, 4)
3 >>> Fraction(3, 9) # phân số sẽ được tối giản nếu có thể
4 Fraction(1, 3)
5 >>> type(Fraction(3, 4)) # các phân số thuộc lớp Fraction
6 <class 'fractions.Fraction'>
7
```

Kiểu dữ liệu

Kiểu số phức

- **Số phức** gồm 2 thành phần: **<Phần thực> + <Phần ảo> j**
- Trong đó
 - <Phần thực> <Phần ảo> là số thực
 - j là đơn vị ảo trong toán học với $j^2 = -1$
- **Tạo một số thực:** **complex(<Phần_thực>,<Phần_ảo>)**
- **Gán giá trị số phức cho 1 biến:**
 - <tên_biến> = <Phần_thực> + <Phần_Ảo>j
- **Để xuất ra phần thực:**
 - <tên_biến>.real
- **Để xuất ra phần ảo:**
 - <tên_biến>.imag

Ví dụ: Nhập một số số phức sau

1. $1 + 3j$
2. Gán biến c có giá trị $2+1j$. Xuất ra phần thực và phần ảo của biến c.
3. $4 + j$ (sẽ có lỗi vì kiểu dữ liệu nhập vào không đúng).
4. Tạo số phức có phần thực là 3, phần ảo là 1.
5. Tạo số phức chỉ có phần thực là 2.
6. Xuất ra kiểu dữ liệu của số $3+1j$.

Kiểu dữ liệu

Kiểu chuỗi

- Dữ liệu kiểu chuỗi rất quan trọng trong lập trình python và trong các vấn đề của khoa học dữ liệu
- Khai báo dữ liệu kiểu chuỗi có thể nằm bên trong cặp nháy đơn ('), hoặc nháy kép (") hoặc 3 dấu nháy kép liên tiếp (""")

```
name = 'matt'
```

```
# chuỗi trong nó có chứa dấu nháy đơn
```

```
with_quote = "I ain't gonna"
```

```
# chuỗi có nội dung nằm trên 2 dòng
```

```
longer = """This string has  
multiple lines init"""
```

Kiểu dữ liệu

Escape sequence

- Escape sequence là một phương pháp để viết các kí tự đặc biệt trong python
 - Tương tự như các ngôn ngữ lập trình khác

Tên	Kí hiệu	Giải thích
Alert	\a	Phát ra một tiếng bíp
Backspace	\b	Đưa con trỏ về lại một khoảng trắng
Newline	\n	Đưa con trỏ xuống dòng tiếp theo
Horizontal tab	\t	In một horizontal tab
Single quote	\'	In ra kí tự '
Double quote	\"	In ra kí tự "
Backslash	\\	In ra kí tự \



Kiểu dữ liệu

Chuỗi trần

- Đặt vấn đề: bạn thao tác với các đường dẫn file, các chuỗi này sẽ có dạng Ổ_đĩa:\Thư_mục\Thư_mục
 - Nếu tên thư mục bắt đầu với các chữ cái t, n, a, v, b,... và kết hợp với kí tự \. thành escape sequence, điều này có thể gây nhầm lẫn khi viết nội dung các chuỗi
- Python cho phép sử dụng một dạng chuỗi, gọi là chuỗi trần, bằng cách bỏ qua escape sequence (ký tự đặc biệt)
 - Cú pháp: r' nội dung chuỗi'

```
0  >>> a = r'\neu mot ngay'    # chuỗi trần, bỏ qua Escape Sequence \n
1  >>> print(a)
2  \neu mot ngay'
```

Kiểu dữ liệu

Kiểu Boolean

- Thường gọi tắt là kiểu Bool/ kiểu Đúng – Sai/ kiểu Luận lý
- Là kiểu giá trị này chỉ có hai giá trị là đúng (True) và sai (False)

```
boolean.py
1 | print (bool(True))
2 | print (bool(False))
3 | print (bool("text"))
4 | print (bool(""))
5 | print (bool('.'))
6 | print (bool(0))
7 | print (bool())
8 | print (bool(3))
9 | print (bool(None))
```

Output

```
1 | True
2 | False
3 | True
4 | False
5 | True
6 | False
7 | False
8 | True
9 | False
```

Kiểu dữ liệu

Chú thích

- Python sử dụng kí tự # để chú thích các đoạn code
- Tất cả các nội dung sau kí tự # sẽ không được dịch

```
>>> #Toi dang soạn bai giang Python  
...  
>>> #Ban dang xem toi soạn Bai giang  
...  
>>>
```

Nhập, xuất dữ liệu

Xuất dữ liệu

- Sử dụng hàm print để in dữ liệu ra màn hình

```
>>> print(42)
```

```
42
```

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

```
a = 3.564
```

```
>>> print("a = \n", a)
```

```
a = 3.564
```

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

```
>>> print("a","b",sep="")
```

```
ab
```

```
>>> print(192,168,178,42,sep=".") 192.168.178.42
```

```
>>> print("a","b",sep=":-)") a:-)b
```

Nhập, xuất dữ liệu

Nhập dữ liệu

- Sử dụng hàm input để nhập dữ liệu từ bàn phím

```
name = input("What's your name? ")  
print("Nice to meet you " + name + " !")  
age = input("Your age? ")  
print("You are already " + age + " years old, " + name  
+ " !")
```

- Có thể kết hợp chuyển kiểu nếu muốn tương minh

```
age = int(input("Your age? "))  
print("You are already %d years old!", age)
```

LOGO

CẢM ƠN!