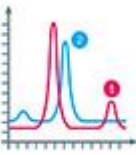


# NHẬP MÔN LẬP TRÌNH KHOA HỌC DỮ LIỆU

---

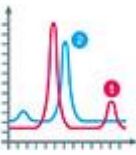
## Bài 2: Ngôn Ngữ Lập Trình Python (1)



# Nội dung

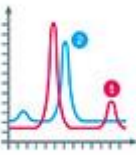
---

1. Giới thiệu ngôn ngữ python
2. Cách thực hiện câu lệnh, chương trình
3. Biến, Kiểu dữ liệu, Khối lệnh
4. Nhập và Xuất dữ liệu
5. Vài ví dụ minh họa



Phần 1

# Giới thiệu ngôn ngữ python



# Giới thiệu ngôn ngữ python

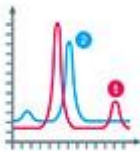
---

- 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

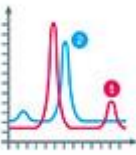


# Giới thiệu ngôn ngữ python

---



- 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)



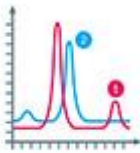
# Giới thiệu ngôn ngữ python

---

- 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)

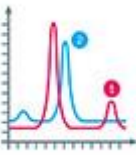
# Ưu điểm của ngôn ngữ python

---



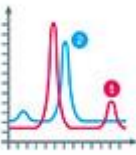
- Có ngữ pháp đơn giản, dễ đọc
- Viết mã ngắn gọn hơn những chương trình tương đương được viết trong C, C++, C#, Java,...
- Có các bộ thư viện chuẩn và các module ngoài, đáp ứng gần như mọi nhu cầu lập trình
- Có khả năng chạy trên nhiều nền tảng (Windows, Linux, Unix, OS/2, Mac, Amiga, máy ảo .NET,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

# Nhưng python cũng có 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

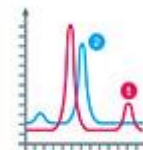




Phần 2

# Cách thực hiện câu lệnh, chương trình

# Cài đặt



Python Software Foundation [US] <https://www.python.org/downloads/>

Python PSF Docs PyPI

python™

About Downloads Documentation Community S

## Download the latest version for Windows

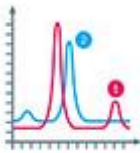
[Download Python 3.6.5](#) [Download Python 2.7.14](#)

Wondering which version to use? [Here's more about the difference between Python 2 and 3.](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#)

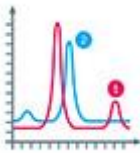
# Khởi chạy



- 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 =
```

# Khởi chạy

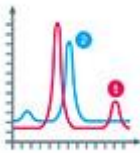


- Chế độ dòng lệnh: “python”
  - Lúc này trình thông dịch python sẽ chờ người dùng gõ từng dòng lệnh
  - Gõ dòng lệnh nào xong, python chạy liền dòng đó
  - Chấm dứt chế độ này bằng cách gõ lệnh: “quit()”

```
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>
```

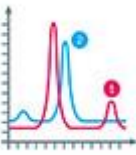
# Soạn thảo mã python



- 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
- ...

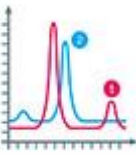
```
C:\DNN\DNN\lesson3a.py (DNN) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  ▾ DNN
    /* lesson2a.py
    /* lesson2b.py
    /* lesson2c.py
    /* lesson2d.py
    /* lesson2f.py
    /* lesson3a.py
lesson3a.py
1 import matplotlib.image as mpimg
2
3 # First, 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()
Line 6, Column 1
Tab Size: 4 Python
```



# Biên dịch mã python

---

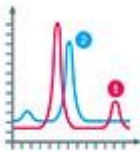
- Trường hợp cần thiết, mã python có thể được biên dịch, kết quả dịch là chương trình dạng bytecode cho máy ảo python
  - Tương tự như trường hợp của ngôn ngữ java
- Mã lệnh dịch được lưu vào file với đuôi `.pyc`
- Việc biên dịch có nhiều lợi điểm, chẳng hạn như khi sử dụng câu lệnh `import` một thư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



Phần 3

# Biến, Kiểu dữ liệu, Khối lệnh

# Biến

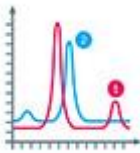


- Biến = vùng bộ nhớ được đặt tên (để dễ thao tác)
- Biến trong python:
  - Có tên, phân biệt chữ hoa/thường
  - Không cần khai báo trước
  - Không cần chỉ ra kiểu dữ liệu
  - Có thể thay đổi sang kiểu dữ liệu khác
  - Nên gán giá trị ngay khi bắt đầu xuất hiện
- Ví dụ:

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



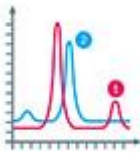
# Biến



- 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)
>>>
```

# 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 (""")

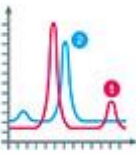
```
name = 'matt'
```

```
# 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 in it"""
```



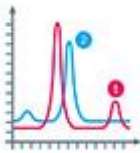
# Escape sequence

- 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 \                       |



# Chuỗi trần

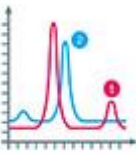


- Đặt vấn đề: bạn thao tác với các đường dẫn file, các chuỗi này sẽ có dạng Ổ\_đĩa:\Thư\_mục\Thư\_mục
  - Nếu tên thư mục bắt đầu với các chữ cái t, n, a, v, b,... và kết hợp với kí tự \. thành escape sequence, điều này có thể gây nhầm lẫn khi viết nội dung các chuỗi
- Python cho phép sử dụng một dạng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

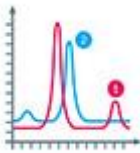
# Chú thích (comment)

---



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

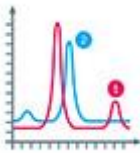
```
>>> #Toi dang soan bai giang Python
...
>>> #Ban dang xem toi soan Bai giang
...
>>>
```



Phần 4

# Nhập và 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="")
```

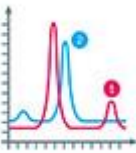
```
ab
```

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

```
192.168.178.42
```

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

```
a:-)b
```



# Nhập dữ liệu

---

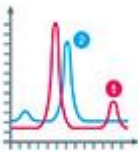
- Sử dụng hàm input để nhập dữ liệu từ bàn phím

```
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)
```

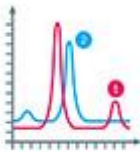




Phần 5

# Vài ví dụ minh họa

# Giải phương trình bậc 2



```
a = float(input("A = "))  
b = float(input("B = "))  
c = float(input("C = "))  
delta = b*b-4*a*c
```

Nhập a,b,c kiểu số thực và tính delta

```
if delta==0:  
    print("Nghiem kep: x = ", str(-b/2/a))
```

Biện luận các trường hợp của delta

```
if delta<0:  
    print("Phuong trinh vo nghiem")
```

Các khối lệnh con được viết thụt vào so với khối cha

```
if delta>0:  
    print("X1 = " + str((-b+delta**0.5)/2/a))  
    print("X2 = " + str((-b-delta**0.5)/2/a))
```

Tính căn bậc 2 bằng phép lũy thừa 0.5