



# Bài 2: Cơ bản về Python

---

TS. Trịnh Tuấn Đạt  
Viện CNTT-TT, ĐHBK Hà Nội



# Nội dung

---

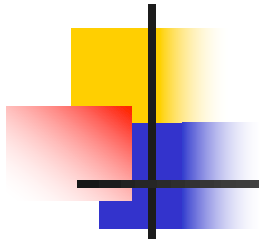
1. Giới thiệu về Python
2. Cú pháp Python cơ bản



# Nội dung

---

1. **Giới thiệu về Python**
2. Cú pháp Python cơ bản



# 1. Giới thiệu về Python

---

- Tính chất:
  - Ngôn ngữ lập trình bậc cao
  - Thông dịch
  - Hướng đối tượng
  - Dễ học, dễ hiểu, dễ bảo trì
  - Nhiều thư viện hỗ trợ
  - Mã nguồn mở (GNU General Public License - GPL)
  - Dễ tích hợp với C, C++, COM, ActiveX, CORBA, và Java.
  - Chạy được trên đa nền tảng
- Guido Van Rossum là người sáng lập ra Python (1985-1990)
- Python 3.0 phát hành 2008



# Cài đặt Python

---

- Link tải:
  - <https://www.python.org/downloads/>
  - Có bản 32 bit, 64 bit cho các nền tảng
- Cài đặt như phần mềm thông thường
- Biến môi trường PATH (Unix) và Path (Window)
  - Là một **xâu ký tự**
  - Giúp hệ điều hành tìm được vị trí các ứng dụng và chương trình thực thi
  - Nếu không thiết lập, phải nhập đường dẫn đầy đủ tới ứng dụng
  - Ví dụ:
    - Windows: `C:\Python`
    - Unix/Linux: `/usr/local/bin/python3`



# Thực thi chương trình Python

## 2 chế độ (mode) lập trình

- Lập trình tương tác (Interactive Mode Programming)

On Linux:

```
$ python
```

```
Python 3.3.2 (default, Dec 10 2013, 11:35:01)
```

```
[GCC 4.6.3] on Linux
```

```
Type "help", "copyright", "credits", or "license" for more information.
```

```
>>>
```

On Windows:

```
C:\Python
```

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32  
bit (Intel)] on win32
```

```
Type "copyright", "credits" or "license()" for more information.
```

```
>>>
```

```
>>> print ("Hello, Python!")
```

- Lập trình kịch bản (Script Mode Programming)

```
$ python test.py
```



# Nội dung

---

1. Giới thiệu về Python
2. **Cú pháp Python cơ bản**

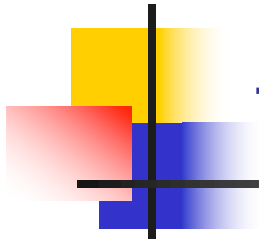


## 2. Cú pháp Python cơ bản

---

- Định danh
- Dòng lệnh và thụt dòng lệnh
- Một lệnh trên nhiều dòng
- Nhiều lệnh trên một dòng
- Hằng xâu
- Chú thích

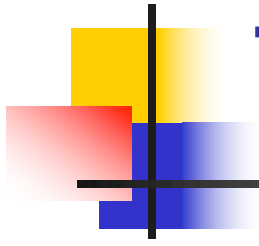




# Định danh trong Python (Identifiers)

---

- Định danh: **tên** của biến, hàm, lớp, module, đối tượng
- Định danh:
  - bắt đầu là chữ cái (A-Z, a-z) hoặc dấu gạch dưới (\_)
  - theo sau bởi 0 hoặc nhiều chữ cái, dấu gạch dưới hoặc các chữ số (từ 0 tới 9)
- Không dùng ký tự đặc biệt (@, \$, và %) trong định danh
- Phân biệt hoa thường (samsung, Samsung, samSung là khác nhau)



# Từ khóa trong Python

<b>and</b>	<b>exec</b>	<b>not</b>
<b>as</b>	<b>finally</b>	<b>or</b>
<b>assert</b>	<b>for</b>	<b>pass</b>
<b>break</b>	<b>from</b>	<b>print</b>
<b>class</b>	<b>global</b>	<b>raise</b>
<b>continue</b>	<b>if</b>	<b>return</b>
<b>def</b>	<b>import</b>	<b>try</b>
<b>del</b>	<b>in</b>	<b>while</b>
<b>elif</b>	<b>is</b>	<b>with</b>
<b>else</b>	<b>lambda</b>	<b>yield</b>
<b>except</b>		



## Dòng lệnh và thụt dòng lệnh trong Python

- Python không dùng {} để tạo khối lệnh, mà dùng **thụt dòng** (indentation)
- Số khoảng trống khi thụt dòng là tùy ý, nhưng các lệnh cùng một khối phải cùng số khoảng trống

```
if True:
    print ("True")
else:
    print ("False")
```

**OK**

```
if True:
    print ("Answer")
    print ("True")
else:
    print "(Answer)"
    print ("False")
```

**Error**

# Một lệnh trên nhiều dòng

- Thông thường, mỗi lệnh chỉ trên một dòng

```
s = "abc" + "def"  
print(s)
```

OK

```
s = "abc" +  
"def"  
print(s)
```

Error

- Để một lệnh trên nhiều dòng, dùng ký tự \

```
s = "abc"\  
+ "def"  
print(s)
```

- Lệnh trong các dấu ngoặc [], {}, hoặc () không cần sử dụng ký tự \

```
days = ['Monday', 'Tuesday',  
        'Wednesday',  
        'Thursday', 'Friday']  
print(days)
```



# Nhiều lệnh trên một dòng

---

- Một dòng có thể có nhiều lệnh, nhưng phải dùng dấu ; để ngăn cách các lệnh

```
x = "Hello Samsung!"; print(x) OK
```

```
x = "Hello Samsung!" print(x) Error
```



# Hằng xâu (chuỗi)

---

- Trong Python, có thể biểu diễn hằng xâu qua dấu nháy đơn ('), nháy kép (") và trích dẫn tam (""" hoặc """"), miễn là trong hằng xâu, dùng **cùng kiểu mở và đóng**.
- Trích dẫn tam: dùng để biểu diễn hằng xâu trên nhiều dòng

```
word = 'word'  
sentence = "This is a sentence."  
paragraph = """This is a paragraph. It is  
made up of multiple lines and sentences."""
```



# Chú thích trong Python

---

- Hai loại chú thích: đơn dòng và đa dòng
- Đơn dòng
  - dấu # không ở bên trong một hằng xâu nào, bắt đầu một comment đơn dòng.
  - Tất cả ký tự sau dấu # cho đến hết dòng được coi là comment và được trình thông dịch bỏ qua

```
# First comment  
print ("Hello, Python!") # second comment
```

```
# This is a comment.  
# This is a comment, too.  
# This is a comment, too.  
# I said that already.
```



# Chú thích trong Python

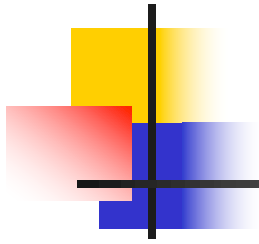
---

- Comment đa dòng: dùng trích dẫn tam

```
#single line comment  
print ("Hello Python")  
"""This is  
multiline comment"""
```

```
'''  
This is a multiline  
comment.  
'''
```

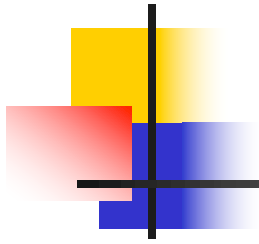




## Dòng trống trong Python (Blank lines)

---

- Dòng trống: dòng chỉ chứa các khoảng trống trắng whitespace (có thể chứa comment)
- Python hoàn toàn bỏ qua dòng trống
- Trong chế độ tương tác, phải nhập một dòng trống để kết thúc một lệnh đa dòng.



# Dừng chương trình, đợi người dùng

- Để dừng chương trình, đợi người dùng

```
input("\n\nPress the enter key to exit.")
```

- Ký hiệu \n: xuống dòng
- Ví dụ áp dụng

```
for i in list(range(20)):  
    print (i)  
input("\n\nPress the enter key to continue.")  
for i in list(range(21, 40)):  
    print (i)
```