

# Giới thiệu ngôn ngữ lập trình Python

---

# Nội dung

---

1. Giới thiệu
2. Môi trường & Công cụ lập trình
3. Cấu trúc một chương trình Python
4. Từ khóa, Biến & Kiểu dữ liệu trong Python
5. Q & A

# 1.1 Giới thiệu về Python

---

- ❖ Python là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object-oriented), và là một ngôn ngữ bậc cao ngữ nghĩa động (dynamic semantics).
- ❖ Python hỗ trợ các module và gói (packages).

## 1.2 Lịch sử

---

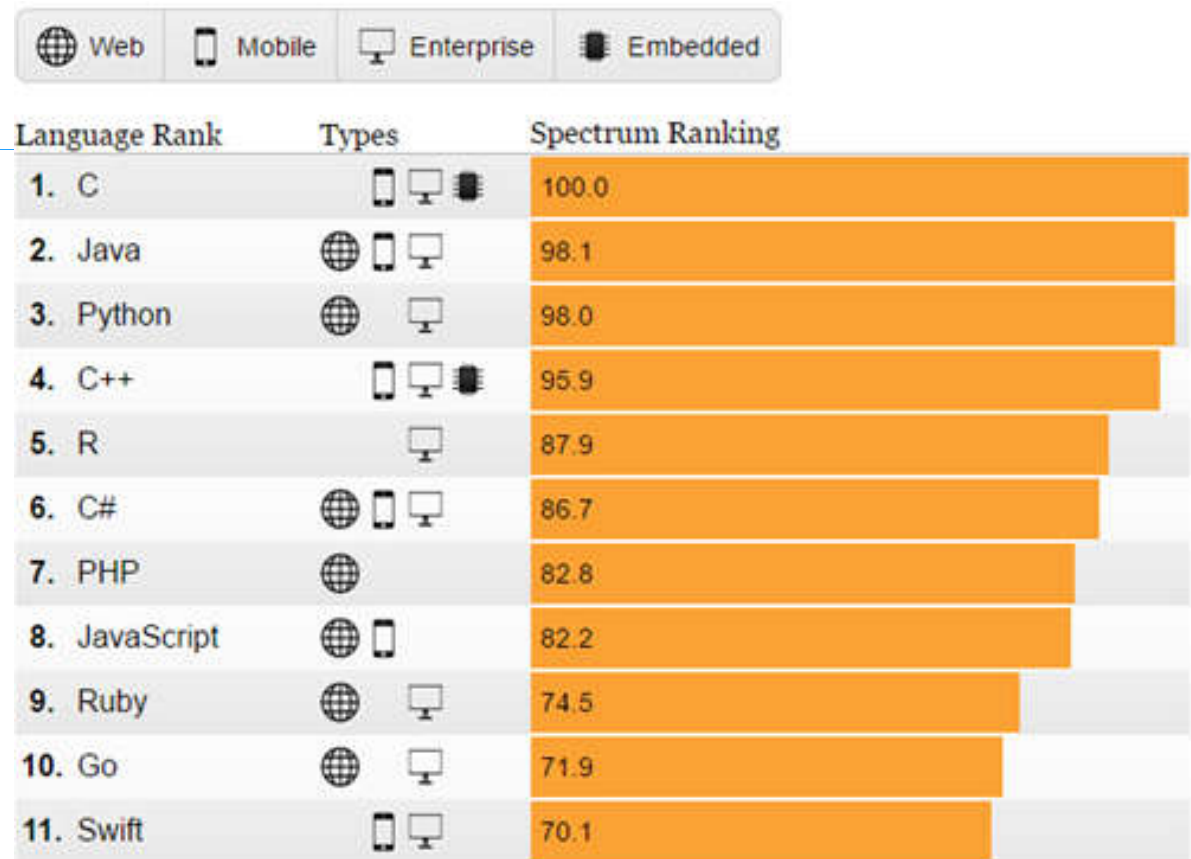
- ❖ **Python** đã được hình thành vào cuối những năm 1980.
- ❖ Ban hành chính thức vào tháng 12 năm 1989 bởi **Guido van Rossum** tại **Centrum Wiskunde & Informatica (CWI)** ở Hà Lan.
- ❖ **Van Rossum** là tác giả chính của **Python**, và ông là một trong những người tiên phong đưa ra hướng phát triển của **Python**.



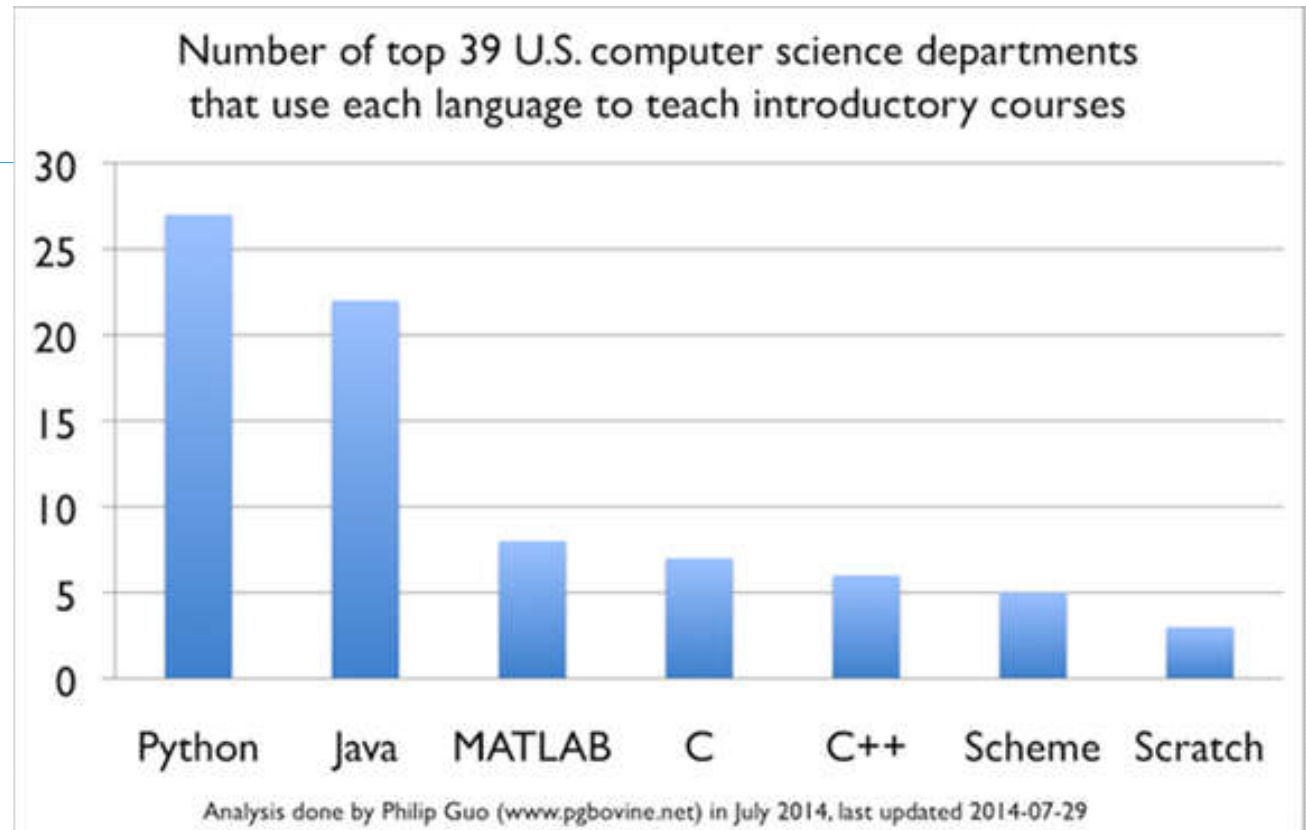
- 
- ❖ Python 2.0 được phát hành vào ngày 16 tháng 10 năm 2000 và có nhiều tính năng mới, trong đó có tính năng hỗ trợ Unicode. Với việc phát hành này quá trình phát triển đã được thay đổi, trở nên minh bạch hơn và tạo cộng đồng lớn.
  - ❖ Python 3.0 được phát hành năm 2008, sau một thời gian dài thử nghiệm.
  - ❖ Cho tới năm 2017, Python đang có phiên bản **3.6.x**.

## 1.3 Hiện tại

❖ Năm 2016 ngôn ngữ Python được xếp hạng thứ 3 trong Top các ngôn ngữ lập trình phổ biến nhất đang được thế giới sử dụng.



❖ Thống kê từ top 39 trường giảng dạy khoa học máy tính thì đa phần các trường sử dụng ngôn ngữ Python để giảng dạy

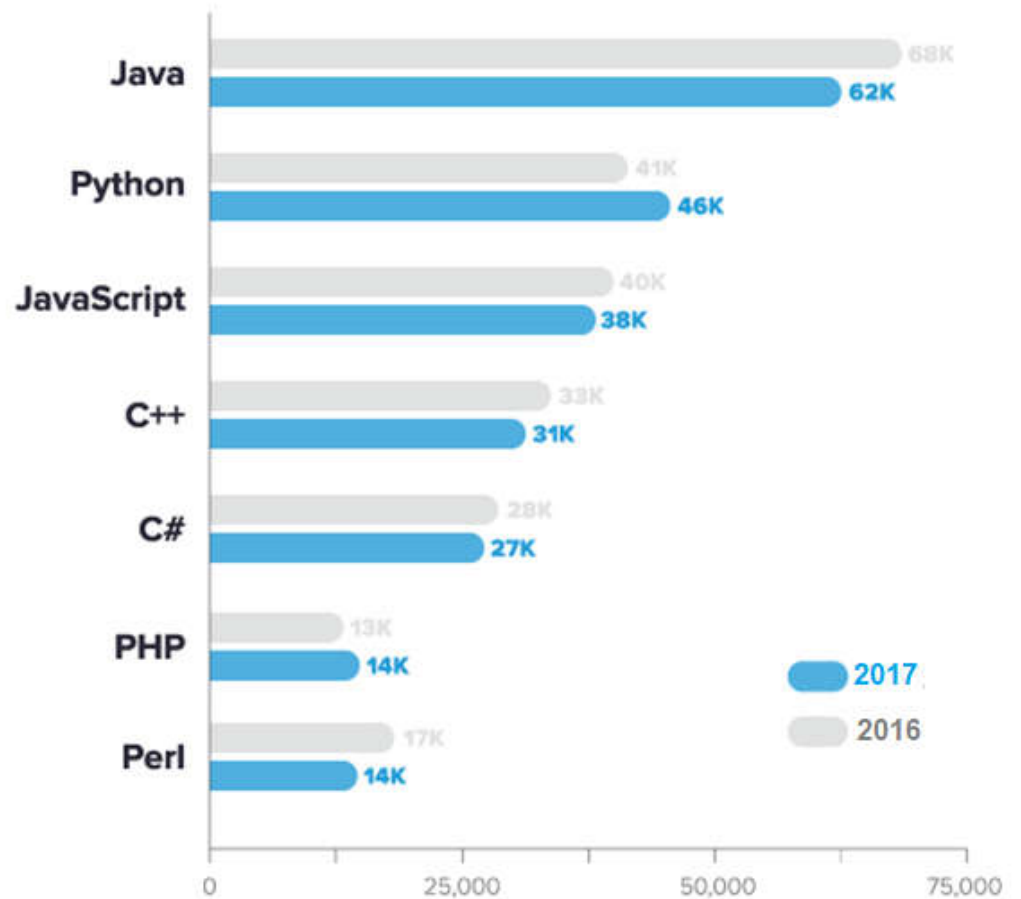


---

❖ Thống kê về số lượng Jobs tuyển dụng trên indeed.com

### Job postings containing top languages

Indeed.com - November, 17th 2017





## 1.4 Đặc điểm (1)

---

- ❖ Python là ngôn ngữ có khả năng chạy trên nhiều nền tảng: Windows, Linux/Unix, Mac và những hệ điều hành khác. Thậm chí có cả những phiên bản được phát triển chạy trên .NET, máy ảo Java. Với cùng một mã nguồn sẽ chạy giống nhau trên mọi nền tảng.
- ❖ Python rất đơn giản và dễ học: Python có cộng đồng lập trình lớn, hệ thống thư viện chuẩn, và cả các thư viện mã nguồn mở được chia sẻ phổ biến trên internet.
- ❖ Python là ngôn ngữ mã nguồn mở được sử dụng và phân tối tự do, ngay cả trong việc thương mại.

## 1.4 Đặc điểm (2)

---

- ❖ Vừa hướng thủ tục (procedural-oriented), vừa hướng đối tượng (object-oriented).
- ❖ Hỗ trợ module và hỗ trợ gói (package).
- ❖ Xử lý lỗi bằng ngoại lệ (Exception).
- ❖ Kiểu dữ liệu động.
- ❖ Có khả năng tương tác với các ngôn ngữ lập trình khác: **C/C++**, **Java**, **.Net** ...

## 2.1 Môi trường & công cụ lập trình

---

### ❖ Các phiên bản:

- 2.x

- 3.x

### ❖ Cài đặt môi trường:

- OS: **Windows** hoặc Ubuntu...

- Python: 2.x hoặc **3.x**

- IDE: **PyCharm** hoặc Eclipse

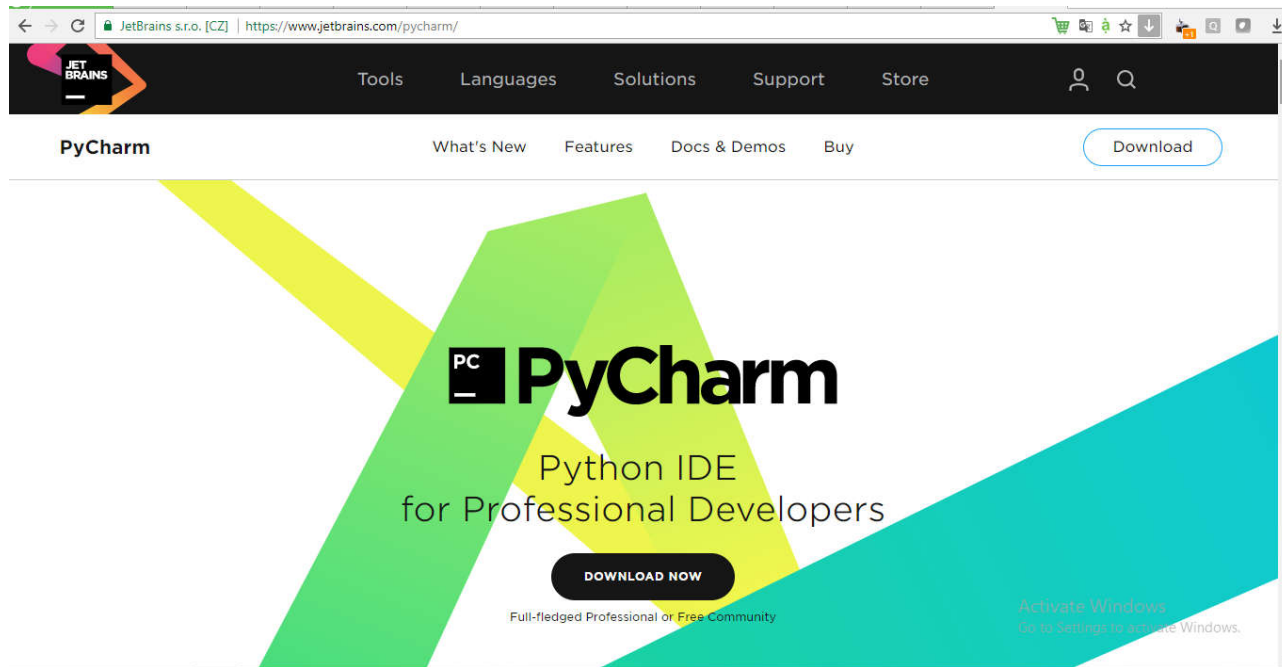
## 2.2 Cài đặt Python

---

- ❖ **Bước 1:** Download python 3.6.5 tại <https://www.python.org/downloads/>
- ❖ **Bước 2:** Cài đặt theo hướng dẫn <http://www.howkteam.vn/course/lap-trinh-python-co-ban/cai-dat-moi-truong-python-1536>

## 2.3 Cài đặt Pycharm

❖ Link download: <https://www.jetbrains.com/pycharm/>



## 3.1 Chương trình đầu tiên

---

### ❖ 'Hello world':

Tạo file main.py.

Gõ lệnh:

```
print('Hello world')
```

Run:

```
F:\sourceCODE\srcPython\2_PYA0418E\basic\venv\Scripts\python.exe F:/sourceCODE/srcPython/2_PYA0418E/basic/main.py
Hello world

Process finished with exit code 0
```

## 3.2 Cú pháp

---

- ❖ Python không cung cấp các dấu ngoặc ôm ({}), 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 có thể khác nhau, nhưng tất cả các lệnh bên trong khối phải được thụt cùng một số lượng khoảng trống như nhau.

---

❖ Code đúng:

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

❖ Code sai:

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



## 4.1 Định danh trong Python

---

- ❖ Một định danh (identifier) trong Python được sử dụng để nhận diện một biến, một hàm, một lớp, hoặc một đối tượng.
- ❖ Một định danh bắt đầu với một chữ cái từ A tới Z hoặc từ a tới z hoặc một dấu gạch dưới (\_) hoặc các chữ số (từ 0 tới 9). Ví dụ: `_a`, `aB`, `Ab...`
- ❖ Python không hỗ trợ các ký tự đặc biệt chẳng hạn như `@`, `$` và `%` bên trong các định danh. Ví dụ như các định danh sau là không phù hợp: `@A`, `%b`, `$c`, `a@`, `_$...`

## 4.1 Từ khóa

---

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

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

## 4.2 Biến

---

- ❖ Biến là các vị trí 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.

## 4.2.1 Gán giá trị cho biến

---

- ❖ Trong Python, không cần khai báo biến một cách tường minh. Phép gán được thực hiện bởi toán tử =.
- ❖ Toán hạng trái của toán tử = là tên biến và toán hạng phải là giá trị được lưu trữ trong biến.

```
a = 10           # Mot phep gan so nguyen
b  = 10.01      # Mot so thuc
name = "Hoang"  # Mot chuoai
print (a)
print (b)
print (name)
```

## 4.2.2 Phép đa gán

---

❖ Python cho phép gán một giá trị đơn cho một số biến đồng thời. Python hỗ trợ hai kiểu đa gán sau:

❑ Gán giá trị đơn cho nhiều biến.

```
a = b = c = 3
```

❑ Gán nhiều giá trị cho nhiều biến (Trong trường hợp này, các giá trị sẽ được gán theo thứ tự mà các biến xuất hiện).

```
one, two, three = 1, 2, 3
```

## 4.3 Kiểu dữ liệu

---

- ❖ Dữ liệu mà được lưu trữ trong bộ nhớ có thể có nhiều kiểu khác nhau. Ví dụ: tuổi của sinh viên được lưu trữ dưới dạng một giá trị số còn địa chỉ được lưu trữ dưới dạng các ký tự chữ-số.
- ❖ Python có nhiều kiểu dữ liệu chuẩn được sử dụng để xác định các hành động có thể xảy ra trên chúng và phương thức lưu trữ cho mỗi kiểu.

---

❖ Python có 5 kiểu dữ liệu chuẩn là:

- ❑ Kiểu **Number**: lưu trữ các giá trị số.
- ❑ Kiểu **String**: là một trong các kiểu phổ biến nhất trong Python dùng lưu trữ chuỗi ký tự. Python coi các lệnh trích dẫn đơn và kép là như nhau.
- ❑ Kiểu **List**: lưu trữ dữ liệu dạng danh sách. Một List trong Python được bao xung quanh bởi các dấu ngoặc vuông [].
- ❑ Kiểu **Tuple**: tương tự list nhưng tuple sử dụng các dấu ngoặc đơn. Các đối tượng trong tuple được phân biệt bởi dấu phẩy và được bao quanh bởi dấu ngoặc đơn () .
- ❑ Kiểu **Dictionary**: Dictionary trong Python là một tập hợp các cặp key và value không có thứ tự. Nó là một khối dữ liệu, được bao quanh bởi các dấu ngoặc móc đơn {}.



# Number

---

- ❖ Khai báo biến kiểu number: `_number = 2, _float=2.01...`
- ❖ Python hỗ trợ 4 kiểu dữ liệu số:
  - ❑ Kiểu **int**: kiểu số nguyên không có dấu thập phân.
  - ❑ Kiểu **long**: là các số nguyên không giới hạn kích cỡ.
  - ❑ Kiểu **float**: số thực có dấu thập phân.
  - ❑ Kiểu **số phức**: là trong dạng  $a + bj$ , với  $a$  và  $b$  là số thực và  $j$  biểu diễn căn bậc hai của  $-1$ . Phần thực là  $a$  và phần ảo là  $b$ . Nói chung, số phức không được sử dụng nhiều trong lập trình Python.

# String

---

❖ Khai báo chuỗi ký tự: `_s1 = 'Hello World! '`; `s2 = "Python Programming" ...`

