

# Chương 5: LẬP TRÌNH PYTHON

**NGUYỄN VĂN HÒA**

**EMAIL: NVHOA@AGU.EDU.VN**

**ref: [cit.ctu.edu.vn/~dtngghi/Python.pdf](http://cit.ctu.edu.vn/~dtngghi/Python.pdf)**

# Nội dung

2

- Giới thiệu
- Cài đặt Python
- Lập trình căn bản

# Giới thiệu Python

3

- Python là ngôn ngữ lập trình cấp cao, tương tác, thông, hướng đối tượng
- Do Guido van Rossum đề xuất từ 1985 – 1990
- M. nguồn Python: giấy phép GNU General Public License (GPL)
- Python sẵn dùng trên Unix, Linux, Mac OS, Windows
- Top 5 ngôn ngữ lập tr. nh phổ biến
- Được sử dụng bởi các tổ chức tập đoàn như Wikipedia, Google, Yahoo,...

# Giới thiệu Python

4

- Dễ học (Easy-to-learn)
- Dễ đọc (Easy-to-read)
- Dễ bảo tr. (Easy-to-maintain)
- Thư viện chuẩn (standard library): tương thích UNIX, Linux, Windows, Mac
- Chế độ tương tác: thực thi, kiểm tra, gỡ rối
- Khả chuyển (Portable): phần cứng, hệ điều hành
- Khả năng mở rộng (Extendable): thêm mô-đun
- Kết nối với các hệ quản trị CSDL
- Lập trình giao diện đồ họa GUI

# Giới thiệu Python

5

- Python được sử dụng trong giảng dạy,
- Tính toán khoa học, Công nghệ sinh-tin học,
- Phát triển ứng dụng Web, Kỹ thuật đồ họa, xử lý ảnh và thị giác máy tính
- Máy học và khai thác dữ liệu, xử lý ngôn ngữ tự nhiên
- Trí tuệ nhân tạo, Lập trình nhúng, quản trị hệ thống

# Cài đặt Python

6

- Windows: cài đặt từ file .msi (python-2.7.5.msi)
- Linux (Ubuntu)
  - Cài từ mã nguồn (Python-2.7.12.tar.xz)
    - ✦ giải nén: `tar -xvf Python-2.7.12.tar.xz`
    - ✦ `./configure --prefix=/home/user/apps/Python`
    - ✦ `make`
    - ✦ `make install`
    - ✦ Cập nhật đường dẫn PATH trong file `~/.profile`
  - Cài từ lệnh apt-get
    - ✦ `sudo apt-get install python2.7 python-pip`

# Lập trình căn bản

7

```
$ python
```

```
Python 2.7.14 (default, Oct 31 2017, 21:12:13)
```

```
[GCC 6.4.0] on cygwin
```

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

```
>>> print "Helloworld"
```

```
Helloworld
```

```
>>>
```

# Soạn thảo và thực thi chương trình

8

- Dùng trình soạn thảo Notepad++, gedit để soạn thảo
- Phần mở rộng của file Python: .py
- Chương trình Hello.py

```
#!/usr/bin/python
print 'Hello World!'
```
- Thực thi

```
$ python Hello.py
Hello World!
```



# Cú pháp: ký tự, từ khóa

9

- Phân biệt ký tự thường hoa
- Các từ khóa của Python được

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

...

# Cú pháp: chú thích, dấu nháy

10

- Sử dụng `#` để chú thích 1 dòng trong chương trình  
`# comment`
- Sử dụng `'''` để chú thích 1 đoạn  
`'''`  
`print("We are in a comment")`  
`print ("We are still in a comment")`  
`'''`
- Dấu `'` và `"`  
`word = 'word'`  
`sentence = "This is a sentence."`  
`message = """This message will`  
`... span several lines."""`

# Cú pháp: khối lệnh

11

- Sử dụng canh lề (bắt buộc) để bao các khối lệnh của hàm, lớp hoặc luồng điều khiển
- Số khoảng trắng dùng để canh lề có thể nhiều ít tùy nhưng tất cả lệnh trong một khối phải được canh lề như nhau

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

# Cú pháp

12

- Lệnh được viết trên nhiều dòng sử dụng ký tự \  
`total = item_one + \  
 item_two + \  
 item_three`
- Lệnh được bao bằng các cặp dấu ngoặc: [], {}, ()  
không cần phải sử dụng ký tự \ để tiếp tục dòng  
`days = ['Monday', 'Tuesday', 'Wednesday',  
 'Thursday', 'Friday']`
- Dấu ; để cách nhiều lệnh trên dòng  
`import sys; x = 'foo'; sys.stdout.write(x + '\n')`

# Cú pháp

13

- Nhóm nhiều câu lệnh đơn tạo nên một khối lệnh và
- cũng được gọi là bộ (suites)
- Các lệnh phức như if, while, def và class cần một dòng header và một bộ
- Dòng header bắt đầu câu lệnh (bằng một từ khoá tương ứng ví dụ như if, def, ...) và kết thúc bằng dấu hai chấm: theo sau là một suite

```
def hi(name):  
    print 'Hello ' + name  
    print 'Have a good day!'  
hi('hoa')
```

# Lệnh **print** trong Python

14

```
>>> print 'Hello'
Hello
>>> print 'One', 'Two'
One Two
>>> a=6.0
>>> b=2.4
>>> name='toto'
>>> print '%d/%f = %f' %(a, b, a/b)
6/2.400000 = 2.500000
>>> print 'Hi %s' %name
Hi toto
>>>
```

# Lệnh **print** trong Python

15

```
>>> a = input("a = ")  
a = 3  
>>> b = input("b = ")  
b = 5.5  
>>> a+b  
8.5  
>>>
```

# Biến, kiểu dữ liệu, phép toán

16

- Tên: ký tự bắt đầu phải là **alphabet** hoặc **\_**
- Không cần khai báo, chỉ gán giá trị (sử dụng dấu =)
- Được tạo ra trong lần đầu gán giá trị
- Phạm vi biến: cục bộ, toàn cục
- Tham khảo đến đối tượng
- Thông tin về kiểu gắn liền với đối tượng
- Kiểu cơ bản: int, float, complex, bool, string
- Các phép toán số học: +, -, \*, /, %, \*\*
- Phép toán so sánh: ==, !=, >, >=, <, <=
- Phép toán luận lý: and, or, not



# Biến, kiểu dữ liệu, phép toán

17

```
>>> item_name='computer'
```

```
>>> item_qty=10
```

```
>>> item_value=1000.5
```

```
>>> print item_name, item_qty,  
item_value
```

```
computer 10 1000.5
```

```
>>> x = y = z = 1
```

```
>>> print x, y, z
```

```
1 1 1
```

```
>>> x,y,z=1,2,'abc'
```

```
>>> print z,y,z
```

```
abc 2 abc
```

```
>>> print x,y,z
```

```
1 2 abc
```

```
>>> x,y = y,x
```

```
>>> print x,y
```

```
2 1
```

# Biến, kiểu dữ liệu, phép toán

18

```
def func1():  
    var1 = "PHP"  
    print "In side func1() var1",var1
```

- ```
def func2():  
    global var1  
    print "In side func2() global var1",var1
```
- ```
func1()
```
- ```
func2()
```

# Biến, kiểu dữ liệu, phép toán

19

```
>>> x=5.5
>>> type(x)
<type 'float'>
>>> x='hello'
>>> type(x)
<type 'str'>
```

```
>>> x=123456
>>> x**2
15241383936
>>> z=3+2j
>>> t=2-1j
>>> z+t
(5+1j)
```

# Biến, kiểu dữ liệu, phép toán

20

```
>>> x=7
```

```
>>> y=2
```

```
>>> x%y
```

```
1
```

```
>>> x/y
```

```
3
```

```
>>> 1.0*x/y
```

```
3.5
```

```
>>> float(x)/y
```

```
3.5
```

```
>>> a=7.2
```

```
>>> int(a)%y
```

```
1
```

```
>>> b=True
```

```
>>> type(b)
```

```
<type 'bool'>
```

# Biến, kiểu dữ liệu, phép toán

21

```
>>> s='Hello world!'
>>> type(s)
<type 'str'>
>>> len(s)
12
>>> print s
Hello world!
>>> print s[0]
H
>>> print s[-1]
!
```

```
>>> print s[2:5]
llo
>>> print s[2]
l
>>> print s[2:]
llo world!
>>> print s*2
Hello world!Hello world!
>>> name = 'Ben ' +
str(10)
>>> print name
Ben 10
```

# Cấu trúc lựa chọn

22

Cú pháp

if (cond1):

...

elif (cond2):

...

else:

...

```
#!/usr/bin/python
```

```
# eq:  $ax + b = 0$ 
```

```
a = input('a = ')
```

```
b = input('b = ')
```

```
if (a==0):
```

```
    if (b==0):
```

```
        print 'PT vo dinh'
```

```
    else:
```

```
        print 'PT vo nghiem'
```

```
else:
```

```
x = -1.0*b/a
```

```
print 'x = ',x
```

# Cấu trúc lặp for

23

Cú pháp:

for iter\_var in sequence:

...

loop\_body

...

```
#!/usr/bin/python
```

```
n = input('n = ')
```

```
s = 0
```

```
for i in range(1, n+1):
```

```
    s = s + i
```

```
print 's = 1 + 2 + ... \n' +  
      '%d = %d' %(n,s)
```

# Lặp với while

24

Cú pháp

while (cond):

...

loop\_body

...

```
#!/usr/bin/python
```

```
n = input('n = ')
```

```
s = 0
```

```
i = 1
```

```
while (i<=n):
```

```
    s = s + i
```

```
    i = i + 1
```

```
print 's = 1 + 2 + ... \n' +  
      + '%d = %d' %(n,s)
```



# Giới thiệu Hàm

25

- Hàm được định nghĩa sẵn

```
>>> import os
```

```
>>> import random as rand
```

```
>>> from math import sqrt, cos, sin
```

```
>>> r = rand.random()
```

```
>>> print r
```

```
0.0256287674608
```

```
>>> x = 64
```

```
>>> sqrt(x)
```

```
8.0
```

```
>>> os.system('ls ./')
```

# Định nghĩa Hàm

26

```
def func_name([parameters])
```

```
...
```

```
    body_of_the func
```

```
...
```

- Tham số tùy chọn, có thể đặt giá trị mặc định
- Hàm có thể hoặc không trả về kết quả hoặc
- Đệ quy

# Định nghĩa Hàm

27

```
#!/usr/bin/python
var1 = "Python"
def func1():
    var1 = "PHP"
    print "In side func1() var1",var1
def func2():
    global var1
    print "In side func2() global var1",var1
func1()
func2()
```

# Định nghĩa Hàm

28

```
#!/usr/bin/python
def addTwo(a,b):
    return a + b
def divide(a,b):
    return a/b, a%b
y = 2
x = 3
z = addTwo(x,y)
print z
p, q = divide(x,y)
print p, q
```

# Định nghĩa Hàm

29

```
#!/usr/bin/python
```

```
def gcd(m,n):
```

```
    while (m != n):
```

```
        if ( m > n):
```

```
            m = m -n
```

```
        else:
```

```
            n = n -m
```

```
    return m
```

```
def coPrime(a,b):
```

```
    if (gcd(a,b) != 1):
```

```
        return
```

```
    else:
```

```
        print "%d and %d are co-prime" %(a,b)
```

```
x, y = 2, 5 ; coPrime(x,y)
```

# Kiểu dữ liệu phức tạp: string

30

```
>>> s = 'a string, with stuff'
```

```
>>> s.count('st')
```

```
2
```

```
>>> s.count('stu')
```

```
1
```

```
>>> s.find('stu')
```

```
15
```

```
>>> s.split(' ')
```

```
['a', 'string,', 'with', 'stuff']
```

```
>>> three = '3'
```

```
>>> three.isdigit()
```

```
True
```

```
>>> supper = s.upper()
```

```
>>> supper
```

```
'A STRING, WITH STUFF'
```

```
>>> s.rjust(30)
```

```
' a string, with stuff'
```

# Kiểu dữ liệu phức tạp: List

31

```
>>> r = [1, 2.0, 4, 5]
```

```
>>> type(r)
```

```
<type 'list'>
```

```
>>> r[1]
```

```
2.0
```

```
>>> r[-1]
```

```
5
```

```
>>> r[1:3]
```

```
[2.0, 4]
```

```
>>> w = r + [10, 19]
```

```
>>> w
```

```
[1, 2.0, 4, 5, 10, 19]
```

```
>>> t = [0.0] * 10
```

```
>>> t
```

```
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

# Kiểu dữ liệu phức tạp: List

32

```
>>> r = [1, 2.0, 4, 5]
>>> r[3] = 'word'
>>> r
[1, 2.0, 4, 'word']
>>> r[0] = [9,8]
>>> r
[[9, 8], 2.0, 4, 'word']
>>> r[0:3] = [1,2,5,6]
>>> r
[1, 2, 5, 6, 'word']
>>> r[1:3] = []
```

```
>>> r
[1, 6, 'word']
>>> len(r)
3
>>> 6 in r
True
>>> r.index(6)
1
```



# Kiểu dữ liệu phức tạp: List

33

```
>>> r = [1, 2.0, 4, 5]
>>> r.append('thing')
>>> r
[1, 2.0, 4, 5, 'thing']
>>> r.append(['another', 'list'])
>>> r
[1, 2.0, 4, 5, 'thing', ['another', 'list']]
>>> r = [1, 2.0, 4, 5]
>>> r.extend(['item', 'another'])
>>> r
[1, 2.0, 4, 5, 'item', 'another']
>>> k = r.pop()
>>> r
[1, 2.0, 4, 5, 'item']
```

# Kiểu dữ liệu phức tạp: List

34

```
>>> w = ['apa', '1', '2', '1234']
```

```
>>> w.sort()
```

```
>>> w
```

```
['1', '1234', '2', 'apa']
```

```
>>> w.reverse()
```

```
>>> w
```

```
['apa', '2', '1234', '1']
```

```
>>> v = w[:]
```

```
>>> v.reverse()
```

```
>>> v
```

```
['1', '1234', '2', 'apa']
```

```
>>> w
```

```
['apa', '2', '1234', '1']
```

# Kiểu dữ liệu hướng đối tượng

35

- Định nghĩa lớp

```
class ClassName:
```

```
    'Optional class documentation string'
```

```
    class_suite
```

- class\_suite: các thuộc tính, phương thức (hàm)
- Tham số đầu tiên của phương thức thường được đặt tên là **self** để thỏa:

```
    obj.meth(args) = class.meth(obj, args)
```

- **name**: public, **\_name**: protected, **\_\_name**: private

# Kiểu dữ liệu hướng đối tượng

36

```
#!/usr/bin/python
```

```
class Employee:
```

```
    'Common base class for all employees'
```

```
    empCount = 0
```

```
def __init__(self, name, salary):
```

```
    self.name = name
```

```
    self.salary = salary
```

```
    Employee.empCount += 1
```

```
def displayCount(self):
```

```
    print "Total Employee %d" %Employee.empCount
```

```
def displayEmployee(self):
```

```
    print "Name: ", self.name, ", Salary: ", self.salary
```

# Vào/ra tập tin

37

- Hàm

```
file_obj = open(file_name [, access_mode][, buffering])  
file_obj.close()  
file_obj.write(string);  
file_obj.read([count]);  
file_obj.tell()  
file_obj.seek(offset[, from])
```

```
import os
```

```
os.rename(current_file_name, new_file_name)  
os.remove(file_name)  
os.mkdir("newdir")  
os.chdir("newdir")  
os.getcwd()  
os.rmdir('dirname')
```

# Vào/ra tập tin

38

```
>>> fo = open("foo.txt", "wb")
>>> print "Name of the file: ", fo.name
Name of the file:  foo.txt
>>> print "Closed or not: ", fo.closed
Closed or not:  False
>>> print "Opening mode: ", fo.mode
Opening mode:  wb
>>> print "Softspace flag: ", fo.softspace
Softspace flag:  0
>>> fo.write("Python is a great language.\nYeah
its great!!\n")
>>> fo.close()
```

# Vào/ra tập tin

39

```
>>> fo=open("foo.txt", "r+")
>>> str = fo.read(10)
>>> str
'Python is '
>>> pos = fo.tell()
>>> pos
10
>>> fo.seek(0)
>>> str = fo.read()
>>> print str
Python is a great language.
Yeah its great!!
```

# Vào/ra tập tin: ví dụ

40

- Đọc file CSV
- Tạo câu lệnh sql để chèn dữ liệu từ file CSV