



Python Basics

Cài đặt và lập trình căn bản
cuong@techmaster.vn

Công cụ cần cài đặt



1. Python 3.7.3
2. PyCharm Community Edition
3. Terminal
 - a. MacOSX: nên dùng iTerm2
 - b. Windows: cài VirtualBox/VMWare Workstation sau đó cài Zorin Lite
 - c. Nếu cài Linux thì nên chọn Ubuntu hoặc Zorin Lite

Có thể lập trình Python trực tuyến



<https://repl.it/languages/python3>

https://www.onlinegdb.com/online_python_compiler

Hỏi đáp về Python #1



Q: Tại sao cần thành thạo Terminal khi lập trình Python?

A: Chúng ta thường khởi động/chạy Python script từ Terminal?

Q: Tại sao nên dùng Linux, Mac OSX để lập trình Python?

A: Bạn vẫn có thể dùng Windows để lập trình Python, tuy nhiên Python thường được dùng để viết script tự động chạy trên máy chủ Linux, Unix. Việc cài đặt lập trình Python trên Mac OSX, Linux cũng dễ dàng

Hỏi đáp về Python #2



Q: Python có thể dùng để lập trình ứng dụng di động?

A: Hiện chưa thể !

Q: Python có thể lập trình ứng dụng web?

A: Python lập trình ứng dụng web động rất tốt

Q: Python có thể lập trình ứng dụng game?

A: Chỉ lập trình ứng dụng game đơn giản để giáo dục kỹ năng lập trình
Chứ không bán được game code bằng Python

Hỏi đáp về Python #3



Q: Thế mạnh của Python là gì?

A: Dễ học, phổ biến, mọi nơi. Ứng dụng xử lý dữ liệu, BigData, Machine Learning, tự động hóa công việc rất tốt

Q: So sánh Python với Java, C#, C++, Golang?

A: Python là ngôn ngữ lập trình scripting, gõ đến đâu, chạy đến đó
Java, C#, C++, Golang là ngôn ngữ lập trình biên dịch

Python cùng loại với Ruby, PHP, JavaScript

Hỏi đáp về Python #4



Q: Máy tôi chỉ có Python 2.7 và Python 3.5 vậy có ổn không?

A: Bạn nên cài đặt Python 3.7.3 hoặc mới hơn để tận dụng hết các tính năng mới?

Q: Tôi sẽ làm được gì với Python sau khi tốt nghiệp lớp học này?

A: Bạn có thể làm ngay một số công việc xử lý dữ liệu, giải bài tập toán...

**Python có thể cài đặt và
chạy trên Windows, Mac,
Linux**

—

Cài đặt Python3 trên MacOSX



1. Cài đặt HomeBrew <https://brew.sh/>
2. Gõ lệnh > brew install python3
3. Sau đó thử gõ > python3 để khởi động python3 và kiểm tra version
4. Cài đặt Pycharm Community Edition

Cài đặt Python3 trên Ubuntu



Tham khảo hướng dẫn này

<https://linuxize.com/post/how-to-install-python-3-7-on-ubuntu-18-04/>

Cài đặt Python3 trên Windows



Tham khảo video này

<https://www.youtube.com/watch?v=OV9WITd9a2U>

Học lệnh Linux trong 15 phút

Nếu bạn không có máy tính cài Linux



- Thử học khóa này <https://www.katacoda.com/kulinacs/courses/linux>
- Hoặc nghịch Ubuntu playground <https://www.katacoda.com/courses/ubuntu/playground>
- Cài Virtual Box sau đó cài Xubuntu ảo hóa bên trong

Linux #1



In ra tài khoản người dùng đang login

whoami

In thư mục hiện tại, pwd = **P**rint **W**orking **D**irectory

pwd

Liệt kê nội dung thư mục, ls = **L**ist directory

ls

Tạo thư mục mới, mkdir = **M**ake a **d**irectory

mkdir ten_thu_muc

Linux #2



Chuyển đến thư mục, **C**hange to **d**irectory
cd duong_dan_thu_muc_moi

Chuyển đến home directory
cd ~

Chuyển ra ngoài một cấp
cd ..

Chuyển ra thư mục gốc
cd /

Linux #3



Copy file hoặc thư mục

cp source target

```
touch demo.md
```

```
echo 'Hello World' >> demo.md
```

```
mkdir LearnPython
```

```
cp demo.md LearnPython/
```

```
cat LearnPython/demo.md
```


Linux #4 quản lý gói phần mềm

```
$ sudo apt list
```

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

```
$ sudo apt install htop
```

```
$ sudo apt remove htop
```

Linux #5



```
$ which python3
```

```
/usr/local/bin/python3
```

```
$ readlink -n `which python3`
```

```
../Cellar/python/3.7.3/bin/python3
```

Linux #6 edit file



```
$ nano demo.py
```

Gõ code vào demo.py sau đó Ctrl+X rồi gõ Y để lưu trước khi thoát

Tiếp đến chạy file demo.py

```
$ python3 demo.py
```

Gõ lệnh Python trong terminal

  ~  python3

Python 3.7.3 (default, Mar 27 2019, 09:23:39)

[Clang 10.0.0 (clang-1000.11.45.5)] on darwin

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

>>> print("Hello")

Hello

>>> msg = "Hi"

>>> print(msg)

Hi

>>> msg = 10

>>> msg += 1

>>> print(msg)

11

```
>>> msg = "Hello"
```

```
>>> print(msg)
```

```
Hello
```

```
>>> a = 10
```

```
>>> b = 12
```

```
>>> a + b
```

```
22
```

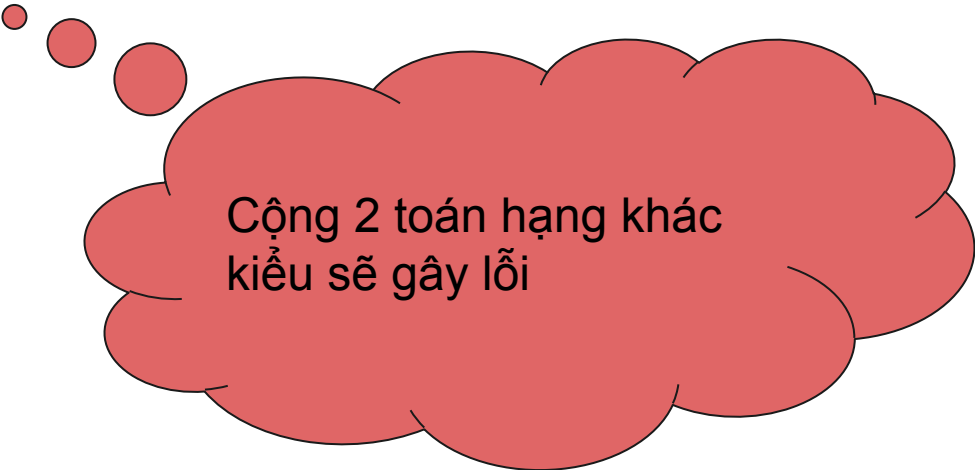
```
>>> c = "string"
```

```
>>> a + c
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: unsupported operand type(s) for +: 'int' and 'str'



Cộng 2 toán hạng khác
kiểu sẽ gây lỗi

Kiểu dữ liệu



```
>>> type('a')  
<class 'str'>
```

```
>>> type('Helo')  
<class 'str'>
```

```
>>> type(10)  
<class 'int'>
```

```
>>> type(3.14)  
<class 'float'>
```

```
>>> type(True)  
<class 'bool'>
```

```
>>> type(False)  
<class 'bool'>
```

```
>>> type([1, 2, 3])  
<class 'list'>
```

```
>>> type(1 + 3j)  
<class 'complex'>
```


Cú pháp căn bản

Comment - ghi chú

Đây là single line comment.

Khi code nên gõ ghi chú để code dễ hiểu.

'''

Đây là multiple lines comment

Khi cần gõ thành nhiều dòng. Nếu viết được bằng tiếng Anh thì tốt nhất.

Khó đâu cứ viết ra. Đừng nhầm trong đầu.

'''

/* Phần quan trọng nhất đó là làm thủ công với mẫu dữ liệu

Chạy thủ công từng bước để tìm ra quy luật

N = 4

0	1	5	6
---	---	---	---

2	4	7	12
---	---	---	----

3	8	11	13
---	---	----	----

9	10	14	15
---	----	----	----

N = 5

0	1	5	6	14
---	---	---	---	----

2	4	7	13	15
---	---	---	----	----

3	8	12	16	21
---	---	----	----	----

9	11	17	20	22
---	----	----	----	----

10	18	19	23	24
----	----	----	----	----

Comment nhiều khi còn chi tiết, kỹ hơn code !

$i + j = k$, $k = 0 \rightarrow 2 * N - 1$

$k < N$

$k \% 2 == 0$, $i = k \rightarrow 0$, $j = 0 \rightarrow k$

$k \% 2 == 1$, $i = 0 \rightarrow k$, $j = k \rightarrow 0$

$k \geq N$

$start = k - N + 1$

$k \% 2 == 0$, $i = (k - start) \rightarrow start$, $j = start \rightarrow (k - start)$

$k \% 2 == 1$, $i = start \rightarrow (k - start)$, $j = (k - start) \rightarrow start$

$k = 6$, $start =$

$k = 7$, $start = 7 - 5 + 1 = 3$, $(3, 4)$, $(4, 3)$

```

k = 0: (0, 0) = 0
k = 1: (0, 1) = 1, (1, 0) = 2
k = 2: (2, 0) = 3, (1, 1) = 4, (0, 2) = 5
k = 3: (0, 3) = 6, (1, 2) = 7, (2, 1) = 8, (3, 0) = 9
k = 4: (4, 0) = 10, (3, 1) = 11, (2, 2) = 12, (1, 3) = 13, (0, 4) = 14, start = 4
-----
k = 5: (1, 4) = 15, (2, 3) = 16, (3, 2) = 17, (4, 1) = 18, start = 1, k-start= 4
k = 6: (4, 2) = 19, (3, 3) = 20, (2, 4) = 21, start = 2, k-start= 4
k = 7: (3, 4) = 22, (4, 3) = 23, start = 3, k-start= 4
k = 8: (4, 4) = 24 start = 4, k-start= 4
*/

```

Viết chức năng cần code
bằng ngôn ngữ tự nhiên,
Bạn sẽ dễ code hơn !

For loop



```
>>> for i in range(1, 5):  
...     print(i)  
...  
1  
2  
3  
4
```

For loop with step



```
>>> for i in range(1, 10, 2):  
...     print(i)
```

1

3

5

7

9

For loop with step down

```
>>> for i in range(10, 1, -2):
```

```
...     print(i)
```

10

8

6

4

2

Loop does not care counter



```
for _ in range(1, 10):  
    print("*")
```



For in collection



```
fruits = ["apple", "banana", "cherry"]  
  
for x in fruits:  
    print(x)
```

String Format



```
>>> name = 'Cuong'
>>> age = 45
>>> print(f"{name} is {age} years old")
Cuong is 45 years old
```

print - in ra màn hình



```
print("In text và xuống dòng")
```

```
print("In text không xuống dòng. ", end = "")
```

```
print("Dòng này nối liền luôn")
```

```
for _ in range(1, 10):
```

```
    print("*", end = " - ")
```

Điều kiện



```
number = 5
```

```
if number > 2:
```

```
    print("Number is bigger than 2.")
```

```
elif number < 2:
```

```
    print("Number is smaller than 2.")
```

```
else:
```

```
    print("Number is 2.")
```

So sánh **is** khác gì **==**

is kiểm tra xem 2 biến có trỏ vào cùng một đối tượng

== kiểm tra 2 biến có thuộc tính bằng nhau không

```
a = [1, 2, 3]
```

```
b = a
```

```
c = [1, 2, 3]
```

a == b	true
a is b	true
a == c	true
a is c	false

Điều kiện inline

n=5

```
result = "Greater than 2" if n > 2 else  
"Smaller than or equal to 2"
```

```
print(result)
```

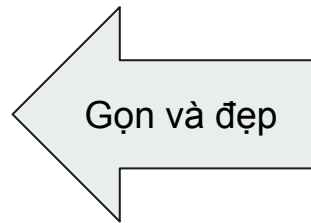
```
message = "Hello" if n > 10 else "Goodbye" if  
n > 5 else "Good day"
```

```
print(message)
```

Điều kiện nối tiếp

```
if n > 2 and n < 6: # Hầu hết các ngôn ngữ khác  
    print("n greater than 2, less than 6")
```

```
if 2 < n < 6: # Đặc sản của Python  
    print("n greater than 2, less than 6")
```



Điều kiện liên hoàn OR

x, y, z = 0, 1, 0

```
if x == 1 or y == 1 or z == 1:  
    print('passed')
```

```
if 1 in (x, y, z):  
    print('passed')
```

```
if x or y or z:  
    print('passed')
```

```
if any((x, y, z)):  
    print('passed')
```


Điều kiện liên hoàn any, all

```
if any((x == 1, y == 1, z == 1)):  
    print('passed')
```

```
if all((x == 1, y == 1, z == 1)):  
    print('passed')  
else:  
    print('not passed')
```

while

Python

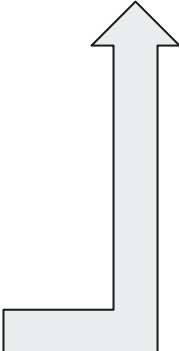
```
i = 0
while i < 20:
    i += 1
    print(i)
```

C

```
for (int i = 0; i < 20; i++)
{
    printf(i);
}
```

break

```
for i in (0, 1, 2, 3, 4):  
    print(i)  
    if i == 2:  
        break
```



0
1
2

continue

```
for i in (0, 1, 2, 3, 4, 5):
```

```
    if i == 2 or i == 4:
```

```
        continue
```

```
    print(i)
```

0
1
3
5

Chạy tiếp, không làm gì

Đảo 2 số không cần biến trung gian



```
a = 10
```

```
b = -10
```

```
a, b = b, a
```

```
print(a, b)
```

Bài tập thực hành



Nhập vào số tự nhiên $N < 10$ ví dụ 5 in ra

*

* *

* * *

* * * *

* * * * *

```
n = int(input("Enter number: "))
```

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

1. Nếu người dùng nhập dữ liệu không đúng làm thế nào?
2. Hãy viết bằng vòng lặp while
3. Thay lệnh '*' * i bằng lệnh for

Xóa phần tử lặp trong mảng



```
>>> m = [1, 2, 4, 1, 4, 6, 7, 8, 9]
```

```
>>> m[0]
```

```
1
```

```
>>> len(m)
```

```
9
```

```
>>> del m[8]
```

```
>>> m
```

```
[1, 2, 4, 1, 4, 6, 7, 8]
```


Hãy thử viết cách khác



```
m = [1, 2, 4, 1, 4, 6, 7, 8, 9]
```

```
mylist = list(dict.fromkeys(m))
```

```
print(mylist)
```

```
[1, 2, 4, 6, 7, 8, 9]
```

import package



Nếu python là cơm, thì package là các món ăn khác nhau.
Python vô dụng nếu không có các package. Vậy package là gì?

- Đóng gói một tập chức năng hữu dụng ví dụ: xử lý ảnh, print PDF, kết nối vào server, mở - đọc - ghi file Excel
- Tái sử dụng code

```
>>> import sys
>>> sys.prefix
'/usr/local/Cellar/python/3.7.3/Frameworks/Python.framework/Versions/3.7'
```

Khi nào gõ lệnh / Khi nào viết script?



Khi cần thử nghiệm nhanh thì gõ lệnh, còn trong hầu hết các trường hợp hãy viết mã vào file có đuôi *.py để lưu trữ, chạy lại dễ dàng, quản lý phiên bản tốt hơn.

Công cụ (IDE: Integrated Development Environment) nào tốt để soạn thảo code Python?

1. Pycharm: chuyên để soạn thảo, gỡ rối, hỗ trợ Python xuất sắc
2. VSCode: đa năng, hỗ trợ lập trình Python
3. Sublime Text: nhẹ, highlight syntax nhưng không debug tốt