



TRƯỜNG ĐẠI HỌC
VĂN LANG

Đạo đức - Ý chí - Sáng tạo

KHOA CÔNG NGHỆ THÔNG TIN

NỀN TẢNG MÁY TÍNH

(Fundamental of computing)

BÀI 3: GIỚI THIỆU NGÔN NGỮ PYTHON



GVGD:

- THS. NGUYỄN THÁI HẢI
hai.nt@vlu.edu.vn
nguyenthaihaivlu@gmail.com
- THS. NGUYỄN THỊ QUYÊN
quyen.nt@vlu.edu.vn



HỌC KỲ II – NĂM HỌC 2020-2021



KHÓA 25Q22, 25Q23



NỘI DUNG

- 01.** Các khái niệm về Python
- 02.** Cài đặt Python
- 03.** Viết chương trình
- 04.** Câu lệnh print
- 05.** Biến và kiểu dữ liệu
- 06.** Các phép toán
- 07.** Độ ưu tiên phép toán
- 08.** Lệnh nhập liệu từ bàn phím
- 09.** Các kiểu xuất dữ liệu



1. Các khái niệm về Python

Python là gì?

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum.



1. Các khái niệm về Python

Tính năng chính của Python

- Ngôn ngữ lập trình đơn giản, dễ học
- Miễn phí, mã nguồn mở
- Khả năng di chuyển
- Khả năng mở rộng và có thể nhúng
- Ngôn ngữ thông dịch cấp cao
- Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến



1. Các khái niệm về Python

Python được dùng ở đâu?

- Lập trình ứng dụng web
- Khoa học và tính toán
- Tạo nguyên mẫu phần mềm
- Ngôn ngữ tốt để dạy lập trình



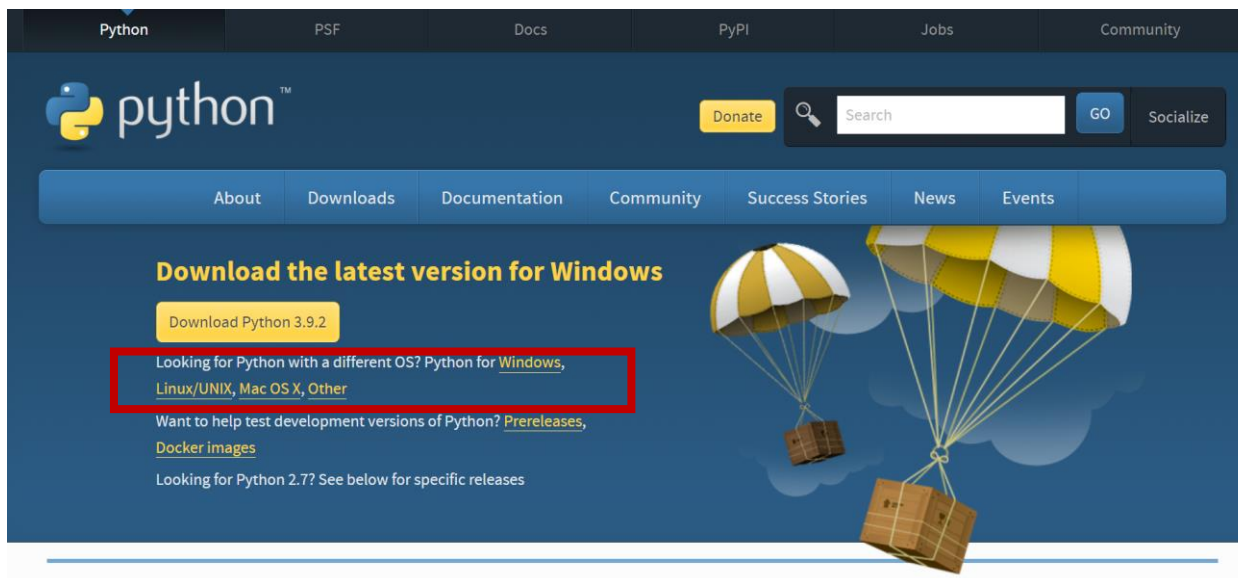
1. Các khái niệm về Python

Lý do để chọn Python

- Cú pháp đơn giản
- Không quá khắt khe
- Viết code ít hơn
- Cộng đồng lớn, hỗ trợ tốt

2. Cài đặt Python

Bước 1: Download Python tại địa chỉ
www.python.org/download

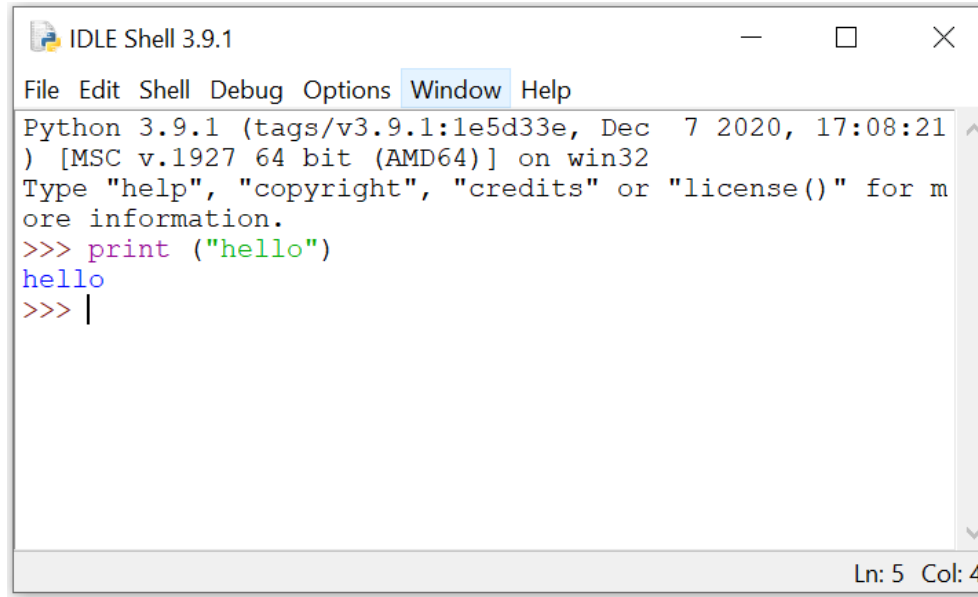


2. Cài đặt Python

Bước 2: Mở ứng dụng Python Shell (vào menu start > python > IDLE Shell 3.9.1)

Dấu >>>: dấu nhắc lệnh chờ thực hiện.

Ví dụ: Sử dụng câu lệnh print để hiển thị chuỗi “hello world”

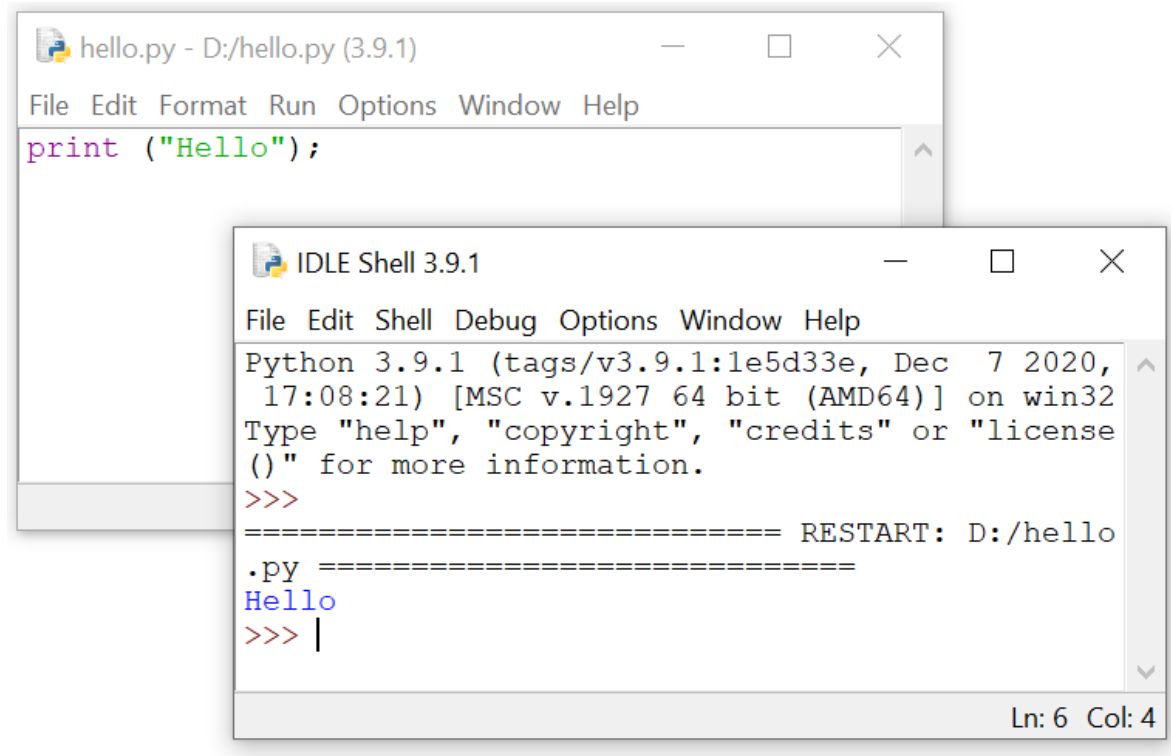


```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("hello")
hello
>>> |
```

Ln: 5 Col: 4

3. Viết chương trình

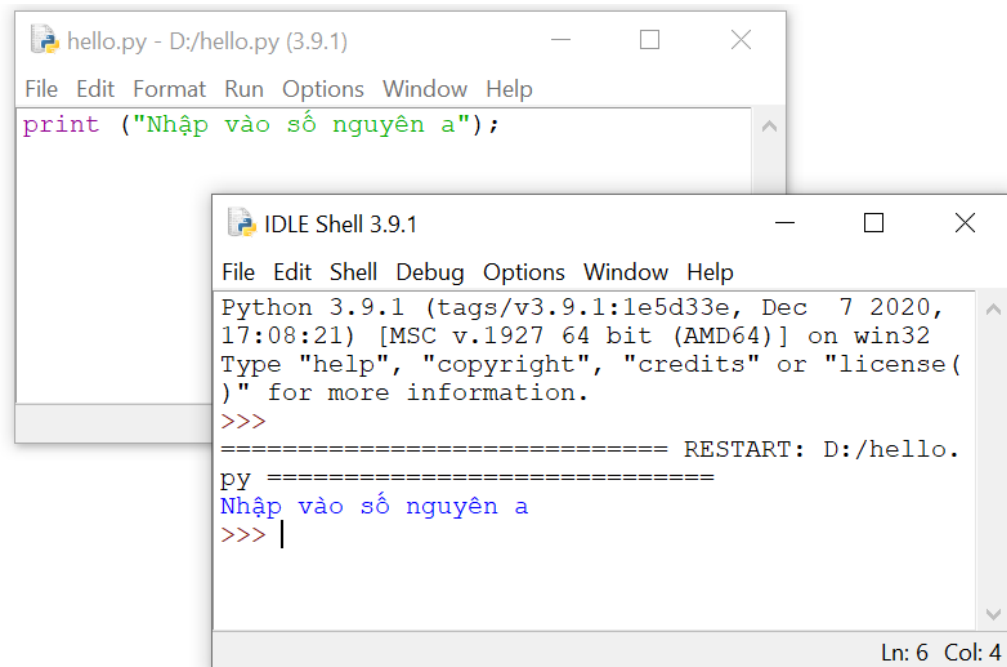
- File -> New File
- Lưu chương trình với định dạng *.Py



The screenshot displays two windows from the Python IDLE 3.9.1 environment. The top window, titled 'hello.py - D:/hello.py (3.9.1)', shows a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code editor contains a single line of Python code: `print ("Hello");`. The bottom window, titled 'IDLE Shell 3.9.1', also has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. It shows the Python 3.9.1 startup message, including the version, build date (Dec 7 2020), and architecture (MSC v.1927 64 bit (AMD64) on win32). It prompts the user to type 'help', 'copyright', 'credits', or 'license' for more information. Below this, the prompt `>>>` is shown, followed by the output of the executed code: `Hello`. The status bar at the bottom right of the shell window indicates 'Ln: 6 Col: 4'.

4. Câu lệnh print

- Được dùng để hiển thị dữ liệu
- Cú pháp 1: **Print** (“chuỗi hiện thị”)
- Ví dụ:



The screenshot displays two windows from the Python IDLE 3.9.1 environment. The top window, titled 'hello.py - D:/hello.py (3.9.1)', contains a single line of Python code: `print ("Nhập vào số nguyên a");`. The bottom window, titled 'IDLE Shell 3.9.1', shows the output of running this script. It displays the Python version and system information, followed by a prompt `>>>`. The output of the `print` statement is shown as `Nhập vào số nguyên a` in blue text. The shell prompt `>>>` is followed by a vertical bar `|`, indicating the next input line.

```
hello.py - D:/hello.py (3.9.1)
File Edit Format Run Options Window Help
print ("Nhập vào số nguyên a");

IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/hello.py =====
Nhập vào số nguyên a
>>> |
```

Ln: 6 Col: 4



4. Câu lệnh print

- Được dùng để hiển thị dữ liệu
- Cú pháp 2: **Print** (“chuỗi hiện thị”, biểu thức/ giá trị, tham số)
- Ví dụ:

```
print ("2 + 2 is", 2 + 2)
print ("3 * 4 is", 3 * 4)
print ("100 - 1 is", 100 - 1)
print ("(33 + 2) / 5 + 11.5 is", (33 + 2) / 5 + 11.5)
```

```
2 + 2 is 4
3 * 4 is 12
100 - 1 is 99
(33 + 2) / 5 + 11.5 is 18.5
```

5. Biến và kiểu dữ liệu

- *Biến*: Biến (variable) là tên được đặt trong bộ nhớ máy tính, người lập trình có thể sử dụng biến để lưu trữ dữ liệu và để truy vấn dữ liệu khi cần thiết.
Ví dụ:

```
a = 1
a = 'Hello World'
a = [1, 2, 3]
a = [1.2, 'Hello', 'W', 2]
```

5. Biến và kiểu dữ liệu

- Hàm `type()` để kiểm tra kiểu dữ liệu của biến
- Ví dụ:

```
x=5
```

```
print (type (x) )
```

 Với `x = 5` ta có kiểu dữ liệu: `<class 'int'>`

```
x='teo'
```

```
print (type (x) )
```

 Với `x = 'teo'` ta có kiểu dữ liệu: `<class 'str'>`

```
x=True
```

```
print (type (x) )
```

 Với `x = True` ta có kiểu dữ liệu: `<class 'bool'>`

```
x=5.5
```

```
print (type (x) )
```

 Với `x = 5.5` ta có kiểu dữ liệu: `<class 'float'>`

```
x=complex (113, 114)
```

```
print (type (x) )
```

 Với `x = complex(113,114)` ta có kiểu dữ liệu: `<class 'complex'>`

5. Biến và kiểu dữ liệu

- **Kiểu dữ liệu:** kiểu dữ liệu (datatype hay type) là một cách phân loại dữ liệu cho trình biên dịch (compiler) hoặc thông dịch (interpreter) hiểu người lập trình muốn sử dụng dữ liệu.

+ **Kiểu số:** Python hỗ trợ hai kiểu số cơ bản: Số nguyên (integer) và số thực (floating point)

Ví dụ: Số nguyên

- a= 5

print(a)

Ví dụ: Số thực

- b= 7.5

print(b);

Ví dụ: Ép kiểu số nguyên sang kiểu số thực

c= 5

c=float(c)

Print(c)

5. Biến và kiểu dữ liệu

+ **Kiểu chuỗi:** Chuỗi là một tập hợp gồm nhiều “ký tự” liên tiếp nhau, được thể hiện/mô tả trong dấu nháy đơn hoặc dấu nháy kép

Ví dụ:

s1= “ Khoa Quản trị kinh doanh”

s2=‘Khoa Công nghệ thông tin’

+ **Kiểu bool:** Kiểu luận lý, để lưu True hoặc False

- Ví dụ 1: t1=True
- Ví dụ 2: t2=False

5. Biến và kiểu dữ liệu

- + **Kiểu complex:** Kiểu số phức,
 - ví dụ1: $z = 2+3j$ thì 2 là phần thực, 3 là phần ảo (j là từ khóa để đánh dấu phần ảo)
 - ví dụ2: $z = \text{complex}(2,3)$ thì 2 là phần thực, 3 là phần ảo
 - khi xuất kết quả ta có thể xuất:
 - `print("Phần thực= ", z.real)` ==> Phần thực= 2
 - `print("Phần ảo= ", z.imag)` ==> Phần ảo= 3



5. Biến và kiểu dữ liệu

Ký tự đặc biệt

Ký hiệu	Ý nghĩa
\n	Dòng mới
\t	Phím tab
\\	\
\'	'
\"	"



5. Biến và kiểu dữ liệu

Xóa một biến

- **Lệnh del:** Xóa 1 biến

Ví dụ:

```
a=7.5
```

```
print (a)
```

```
del a
```

```
print (a)
```

Kết quả

```
7.5
```

```
Traceback (most recent call last):  
  File "D:/a1.py", line 4, in <module>  
    print (a)  
NameError: name 'a' is not defined
```

5. Biến và kiểu dữ liệu

Kiểm tra vùng lưu trữ giá trị của biến

- Kiểm tra vùng lưu trữ giá trị của các biến int, float bằng cách import thư viện sys

```
import sys
print("Thông tin chi tiết của int:")
print(sys.int_info)
print("Thông tin chi tiết của float:")
print(sys.float_info)
```

- Kết quả:

Thông tin chi tiết của int:

```
sys.int_info(bits_per_digit=30, sizeof_digit=4)
```

Thông tin chi tiết của float:

```
sys.float_info(max=1.7976931348623157e+308, max_exp=1024, max_10_exp=308, min
=2.2250738585072014e-308, min_exp=-1021, min_10_exp=-307, dig=15, mant_dig=53
, epsilon=2.220446049250313e-16, radix=2, rounds=1)
```

5. Biến và kiểu dữ liệu

- Ghi chú 1 dòng: Python dùng từ khóa # để cho phép ta ghi chú 1 dòng:

```
GhiChu.py x
1  #đây là ghi chú 1 dòng
2  a=5
3  b=2
4  c=a-b
5  if c<0:
6      print(c)
7  else:
8      print(c*2)
```

5. Biến và kiểu dữ liệu

- Ghi chú nhiều dòng lệnh: 3 cặp nháy đôi `""" """` hoặc 3 cặp nháy đơn `''' '''`

```
"""  
Giải phương trình bậc 1:  $ax+b=0$   
Có 3 trường hợp để biện luận  
Nếu hệ số  $a = 0$  và hệ số  $b = 0 \implies$  vô số nghiệm  
Nếu hệ số  $a = 0$  và hệ số  $b \neq 0 \implies$  vô nghiệm  
Nếu hệ số  $a \neq 0 \implies$  có nghiệm  $-b/a$   
"""  
  
a = 0  
b = 113  
if a == 0 and b == 0:  
    print("Vô số nghiệm")  
elif a == 0 and b != 0:  
    print("Vô nghiệm")  
else:  
    print("Có No X=", -b/a)
```

6. Các phép toán

a. Phép toán số học

Phép toán	Mô tả	Ví dụ
+	Cộng	$12 + 4.9 \Rightarrow$ kết quả 16.9
-	Trừ	$3.98 - 4 \Rightarrow$ kết quả -0.02
*	Nhân	$2 * 3.4 \Rightarrow$ kết quả 6.8
/	Chia	$9 / 2 \Rightarrow$ kết quả 4.5
//	Chia lấy phần nguyên	$9 // 2 \Rightarrow$ kết quả 4
%	Chia lấy phần dư	$9 \% 2 \Rightarrow$ kết quả 1
**	Lũy thừa	$3 ** 4 \Rightarrow$ kết quả 81

6. Các phép toán

b. Phép gán

Phép toán	Mô tả	Ví dụ	Tương đương với
=	Phép gán giá trị bên phải cho biến bên trái dấu bằng	$x=5$	
+=	Cộng và gán	$x=2$ $x+=5$ $\Rightarrow x=7$	$x=x+5$
-=	Trừ và gán	$x=2$ $x-=5$ $\Rightarrow x=-3$	$x=x-5$
=	Nhân và gán	$x=2$ $x=5$ $\Rightarrow x=10$	$x=x*5$

6. Các phép toán

b. Phép gán

Phép toán	Mô tả	Ví dụ	Tương đương với
/=	Chia và gán	$x=7$ $x/=5$ $\Rightarrow x=1.4$	$x=x/5$
//=	Chia và gán (lấy nguyên)	$x=7$ $x//=5$ $\Rightarrow x=1$	$x=x//5$
%=	Chia lấy dư	$x=7$ $x\%=5$ $\Rightarrow x=2$	$x=x\%5$
=	Lấy lũy thừa và gán	$x=2$ $x=3$ $\Rightarrow x$ là 2 mũ 3 $=8$	$x=x**3$

6. Các phép toán

c. Phép toán so sánh

Phép toán	Mô tả	Ví dụ
==	So sánh bằng	5 == 5 => kết quả True
!=	So sánh không bằng	5 != 5 => kết quả False
<	So sánh nhỏ hơn	5 < 5 => kết quả False
<=	So sánh nhỏ hơn hoặc bằng	5 <= 5 => kết quả True
>	So sánh lớn hơn	5 > 5.5 => kết quả False
>=	So sánh lớn hơn hoặc bằng	113 >= 5 => kết quả True
is	Trả về true nếu các biến ở hai bên toán tử cùng trỏ tới một đối tượng(hoặc cùng giá trị), nếu không là false	x=5 y=5 print(x is y) =>kết quả là True
is not	Trả về false nếu các biến ở hai bên toán tử cùng trỏ tới một đối tượng(hoặc cùng giá trị), nếu không là true	x=5 y=5 print(x is not y) =>kết quả là False

6. Các phép toán

d. Phép toán logic

Phép toán	Mô tả	Ví dụ
and	Phép toán Và: Nếu cả hai điều kiện là True thì kết quả sẽ là True	<pre>x=2016 print(x%4==0 and x%100!=0) =>True</pre>
or	Phép toán Hoặc: Chỉ cần một điều kiện True thì nó True, tất cả điều kiện False thì nó False	<pre>x=2016 print((x%4==0 and x%100!=0) or x%400==0) =>True</pre>
not	Phép toán Phủ định. Thông thường nó được dùng để đảo ngược trạng thái logic của toán hạng	<pre>x=4 if (not x>=5): print("Ngắm gà khỏa thân và nài chuỗi") else: print("Đậu")</pre>

7. Độ ưu tiên phép toán

Thứ tự ưu tiên	Phép toán	Miêu tả
1	**	Toán tử mũ
2	* / % //	Phép nhân, chia, lấy phần dư và phép chia lấy phần nguyên
3	+ -	Toán tử Cộng, Trừ
4	<= < > >=	Các toán tử so sánh
5	<> == !=	Các toán tử so sánh
6	= %= /= //=-= += *_=**=	Các toán tử gán
7	is , is not	Các toán tử so sánh
8	not, or, and	Các toán tử Logic



8. Lệnh nhập liệu từ bàn phím

Trong Python để nhập liệu từ bàn phím ta dùng hàm `input()`. Giá trị nhập vào của hàm `input()` thường là kiểu chuỗi, do đó ta cần chuyển kiểu nếu như muốn lưu trữ giá trị nhập vào không phải kiểu chuỗi.

```
print("Mời bạn nhập cái gì đó:")  
s=input()  
print("Bạn nhập:", s)  
print("Kiểu dữ liệu:", type(s))
```



8. Lệnh nhập liệu từ bàn phím

Đưa về số int

```
print("Mời thím nhập int:")  
x=int(input())  
print("Bạn nhập:",x)  
print("Kiểu dữ liệu:",type(x))
```

8. Lệnh nhập liệu từ bàn phím

Đưa về số int

```
print("Mời thím nhập int:")  
x=int(input())  
print("Bạn nhập:", x)  
print("Kiểu dữ liệu:", type(x))
```



8. Lệnh nhập liệu từ bàn phím

Đưa về số float

```
print("Mời thím nhập int:")  
x=float(input())  
print("Bạn nhập:",x)  
print("Kiểu dữ liệu:",type(x))
```

8. Lệnh nhập liệu từ bàn phím

Đưa về số boolean

```
def StrToBool(s):  
    return s.lower() in ("yes", "true", "t", "1")
```

```
print("Mời thím nhập bool:")  
x = StrToBool(input())  
print("Bạn nhập:", x)  
print("Kiểu dữ liệu:", type(x))
```




8. Lệnh nhập liệu từ bàn phím

Ngoài ra hàm input() còn có cho phép ta nhập nhãn tiêu đề vào như sau:

```
x=input("Mời bạn nhập giá trị gì đó:")  
print("Bạn nhập: ",x)
```



9. Các kiểu xuất dữ liệu

Để xuất lặp dữ liệu ra màn hình ta làm như sau:

```
print ('*' * 15)
```

→ Xuất dấu * 15 lần



9. Các kiểu xuất dữ liệu

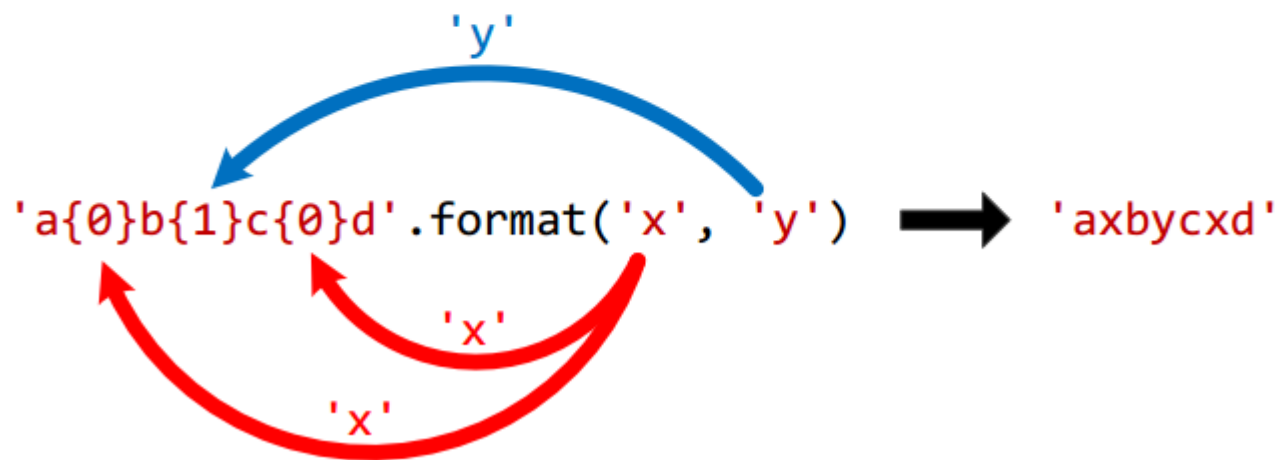
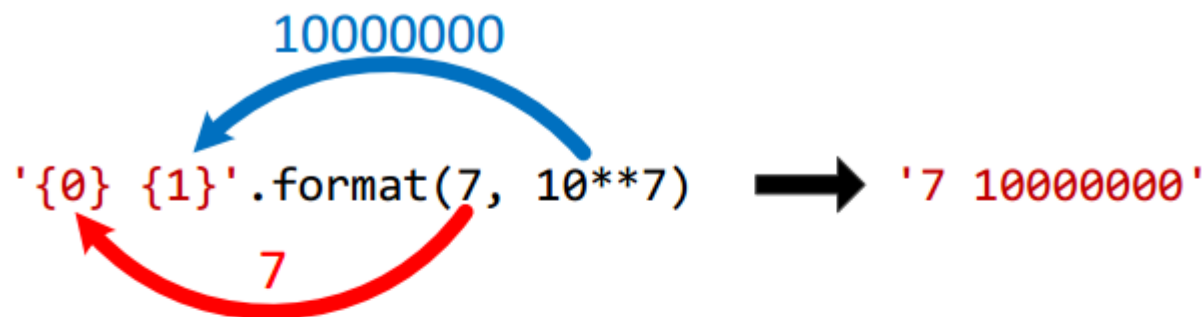
Để xuất lặp dữ liệu ra màn hình ta làm như sau:

```
print ('*' * 15)
```

→ Xuất dấu * 15 lần

9. Các kiểu xuất dữ liệu

Dùng hàm format để xuất dữ liệu `print('{0} {1}'.format(7, 10**7))`



10. Các kiểu xuất dữ liệu

▶ Xuất căn phải ta có thể làm như sau:

```
print('-'*15)
print('{0:>2} {1:>11}'.format('STT', 'Giá trị'))
print('-'*15)
print('{0:>2} {1:>11}'.format(1, 10**10))
print('{0:>2} {1:>11}'.format(2, 10**9))
print('{0:>2} {1:>11}'.format(3, 10**8))
print('{0:>2} {1:>11}'.format(4, 10**7))
print('{0:>2} {1:>11}'.format(5, 10**6))
print('{0:>2} {1:>11}'.format(6, 10**5))
print('{0:>2} {1:>11}'.format(7, 10**4))
print('{0:>2} {1:>11}'.format(8, 10**3))
print('{0:>2} {1:>11}'.format(9, 10**2))
print('{0:>2} {1:>11}'.format(10, 10**1))
print('-'*15)
```

```
-----
STT      Giá trị
-----
1 10000000000
2 1000000000
3 100000000
4 10000000
5 1000000
6 100000
7 10000
8 1000
9 100
10 10
-----
```



