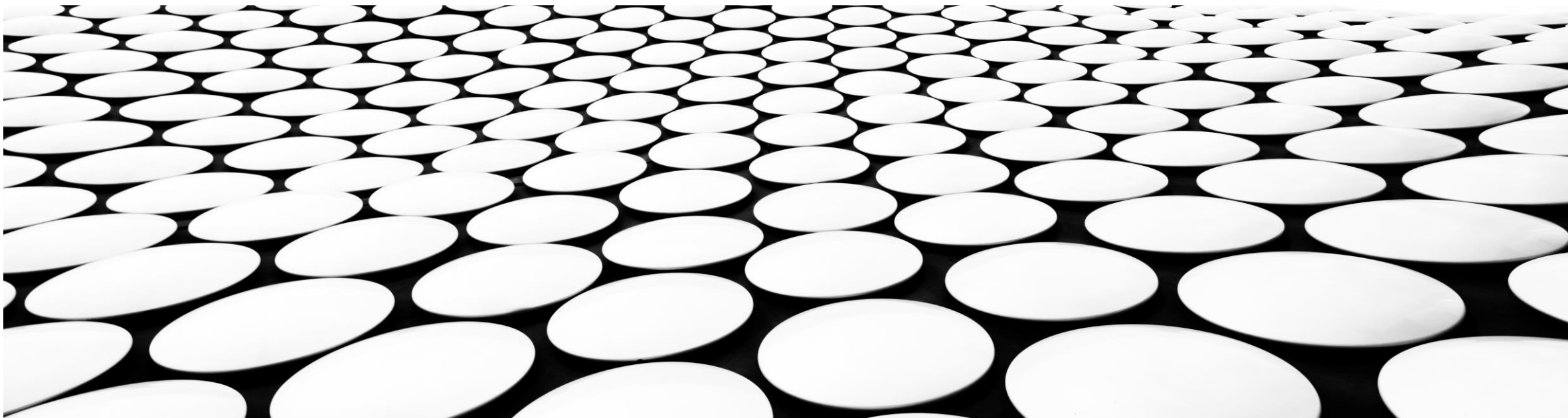

KĨ THUẬT LẬP TRÌNH PYTHON

NGUYỄN MẠNH HÙNG









































CHƯƠNG 1: NHẬP MÔN LẬP TRÌNH PYTHON



1. GIỚI THIỆU VÀ CÀI ĐẶT PYTHON

- **Python** là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do **Guido van Rossum** tạo ra và lần đầu ra mắt vào năm **1991**.
- Python được thiết kế với ưu điểm mạnh là **dễ đọc, dễ học và dễ nhớ**. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Một số ngôn ngữ lập trình thông dụng

Language Rank	Types	Spectrum Ranking
1. C	  	100.0
2. Java	  	98.1
3. Python	 	98.0
4. C++	  	95.9
5. R		87.9
6. C#	  	86.7
7. PHP		82.8
8. JavaScript	 	82.2
9. Ruby	 	74.5
10. Go	 	71.9
11. Swift	 	70.1
12. Arduino		69.9
13. Assembly		68.6
14. Matlab		68.5
15. Scala	 	66.9
16. HTML		65.6
17. Perl	 	58.5
18. Visual Basic		56.8
19. Shell		54.2
20. Objective-C	 	53.4

■ Cài đặt **Jupyter notebook** với **Anaconda**:

- Jupyter notebook là ứng dụng chạy trên nền web cho phép chạy interactive python.
- Anaconda là một công cụ tổng hợp đã tích hợp sẵn rất nhiều các gói phần mềm, thư viện giúp chúng ta có một môi trường để thực hiện xây dựng các ứng dụng Python. Anacoda cũng tích hợp sẵn Jupyter bên trong do đó bạn chỉ cần cài đặt Anacoda là đủ.

Anaconda | Individual Edition

anaconda.com/products/individual

Anaconda Installers

Windows

Python 3.8

- 64-Bit Graphical Installer (466 MB)**
- 32-Bit Graphical Installer (397 MB)

MacOS

Python 3.8

- 64-Bit Graphical Installer (462 MB)
- 64-Bit Command Line Installer (454 MB)

Linux

Python 3.8

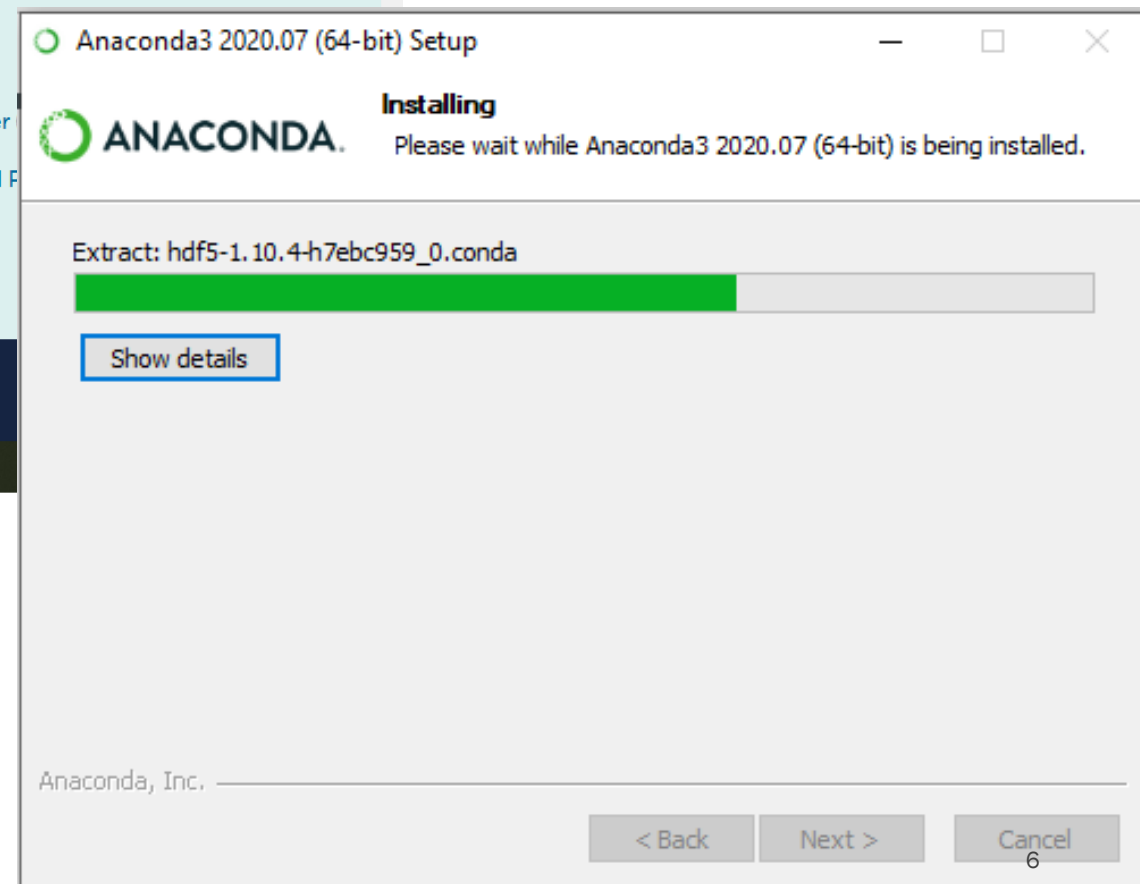
- 64-Bit (x86) Installer
- 64-Bit (Power8 and P
- MB)

This website uses cookies to ensure you get the best experience on our website. [Privacy Policy](#)

https://repo.anaconda.com/archive/Anaconda3-2020.07-Windows-x86_64.exe

2. Cài đặt
Anaconda

1. Tải
Anaconda



3. Mở Anaconda

Home

Environments

Learning

Community

Documentation

Developer Blog



Applications on

base (root)

Channels

Refresh



CMD.exe Prompt

0.1.1

Run a cmd.exe terminal with your current environment from Navigator activated

Launch



JupyterLab

2.1.5

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch



Notebook

6.0.3

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch



Powershell Prompt

0.0.1

Run a Powershell terminal with your current environment from Navigator activated

Launch



Qt Console

4.7.5

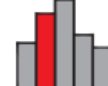
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.



Spyder

4.1.4

Scientific PYTHON Development Environment. Powerful Python IDE with advanced editing, interactive testing,



Glueviz

0.15.2

Multidimensional data visualization across files. Explore relationships within and among related datasets.



Orange 3

3.26.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows

Launching notebook



Type here to search



EN

22:54
17/08/2020

13

Home Page - Select or create a notebook x | Untitled1 - Jupyter Notebook x | +

localhost:8888/tree

jupyter

Files Running Clusters

Select items to perform actions on them.

0 /

3D Objects

anaconda3

Upload New

Python 3

Other:

Text File

Folder

Terminal

Create a new notebook with Python 3

15 hours ago

an hour ago

15 hours ago

15 hours ago

15 hours ago

2 months ago

15 hours ago

15 hours ago

15 hours ago

15 hours ago

Show all

22:56 17/08/2020

5. Mở Python (interactive mode)

Home Page - Select or create a notebook x | Untitled1 - Jupyter Notebook x | +

localhost:8888/notebooks/Untitled1.ipynb?kernel_name=python3

jupyter Untitled1 Last Checkpoint: 2 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

In []:

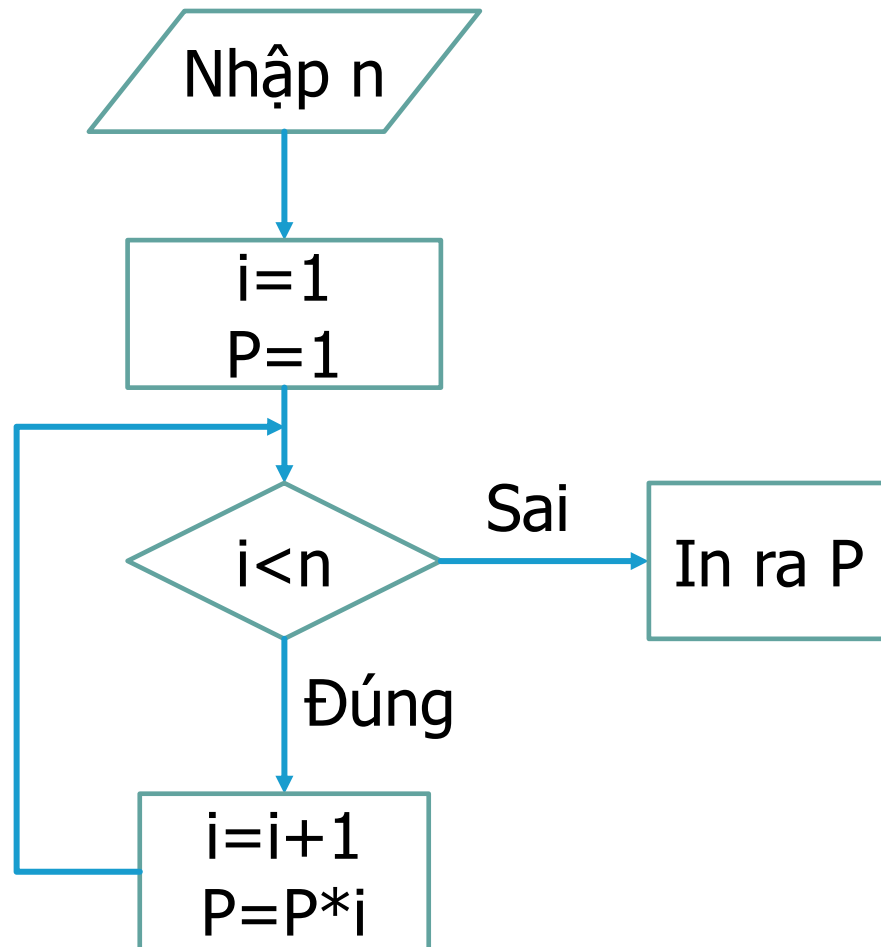
22:57 17/08/2020

8

THẾ NÀO LÀ MỘT CHƯƠNG TRÌNH MÁY TÍNH?

- Khi giải quyết một bài toán bằng máy tính, người ta thường phát triển/xây dựng một **thuật toán**, đó là một danh sách tuần tự các lệnh để giải quyết tất cả các vấn đề của bài toán có thể nảy sinh.
- **Lập trình** là một quá trình thực hiện thuật toán và mã hóa nó thành kí hiệu, ngôn ngữ lập trình, để nó có thể được thực thi bởi máy tính. Không có thuật toán, không có chương trình.
- Một **chương trình máy tính** là một dãy các lệnh điều khiển hoạt động của máy tính.

- Ví dụ: **Nhập** vào số tự nhiên n và **tính giai thừa** $n!$



```
n=int(input("Nhập số n="))
P=1
for i in range(n):
    i=i+1
    P=P*i
print("%d!=%d" % (n,P))
```

Nhập số $n=3$
 $3!=6$

2. VÀO/RA DỮ LIỆU TỪ THIẾT BỊ CHUẨN

1. Đưa kết quả ra màn hình

- ❖ Cú pháp: `print (<danh sách kết quả>)`
- ❖ Tham số: `sep` (mặc định `sep = " "`)
`end` (mặc định `end = "\n"`)

2. Nhập dữ liệu từ bàn phím

Cú pháp: `input()`

```
print("Hello World")  
print("21 + 13 =",21+13)  
print("Chào mừng","đến với","Python")
```

Hello World
21 + 13 = 34
Chào mừng đến với Python

```
print("Hello","World",sep="")  
print("Hello","World",sep="-")
```

HelloWorld
Hello-World

```
print(21, end = "+")  
print(13, end = "=")  
print(21+13)
```

21+13=34

```
x=input("Mời nhập tên của bạn:")  
print("Tên được nhập là:",x)
```

Mời nhập tên của bạn:Nguyễn Văn A
Tên được nhập là: Nguyễn Văn A

Thực hành:

Hãy sử dụng câu lệnh nhập vào họ tên, năm sinh, quê quán, giới tính của mình.
Sau đó in ra lời chúc cho mình tại thời điểm 5 năm tới.

3. BIẾN VÀ GÁN GIÁ TRỊ CHO BIẾN

- **Biến** (variable): Là một đối tượng mà giá trị của nó có thể thay đổi khi thực hiện chương trình. Mỗi biến đều được đặt tên và cần một lượng ô nhớ máy tính tương ứng để lưu giá trị của nó. Trong Python, mỗi biến là một con trỏ chỉ đến ô nhớ chứa giá trị đã được gán cho biến đó.



- Python không có lệnh khai báo biến mà có lệnh gán trực tiếp giá trị cho biến:

<Tên biến> = <Giá trị>

- Để lấy địa chỉ ô nhớ lưu trữ giá trị của một biến, ta dùng hàm **id(Tên biến)**.

```
x=5
print("Giá trị:", x)
print("Địa chỉ ô nhớ: ",id(x))
x="Python cơ bản"
print("Giá trị:", x)
print("Địa chỉ ô nhớ: ",id(x))
```

```
Giá trị: 5
Địa chỉ ô nhớ: 140734656165792
Giá trị: Python cơ bản
Địa chỉ ô nhớ: 2753494481584
```

■ Một số **quy tắc đặt tên** biến:

- ❖ Biến có thể chứa các kí tự chữ cái và chữ số
- ❖ Không chứa kí tự trống “white space” và các kí tự đặc biệt khác như **+** , **-** , ***** , **/** ...
- ❖ Tên biến không thể bắt đầu bằng kí tự số
- ❖ Không sử dụng từ khóa, tên hàm đã được ngôn ngữ lập trình sử dụng, ví dụ như **for**, **while**, **sin**, ...
- ❖ Tên biến nên đặt ngắn gọn, có tính gợi nhớ

- Một số cách **gán rút gọn** giá trị cho biến:

Cách gán thông thường	Cách gán rút gọn
$a = 2, b = 2, c = 2$	$a=b=c=2$
$a = 1, b = 2, c = 3$	$a,b,c=1,2,3$
Tráo đổi giá trị a và b $tem = a, a = b, b = tem$	$a,b = b,a$

4. KIỂU DỮ LIỆU SỐ

- Python hỗ trợ 3 kiểu dữ liệu số:

- ❖ Số nguyên: **int**

- Ví dụ: $x=2$, $y=5$

- ❖ Số thực: **float**

- Ví dụ: $x=2.0$, $\pi=3.14$

- ❖ Số phức: **complex**

- Ví dụ: $x=2-5j$, $y=\text{complex}(4,7)$

- Để xác định kiểu dữ liệu, dùng hàm:

- type(<tên biến>)**

Phép toán	Kí hiệu	Ví dụ
Cộng	+	$4+3=7$
Trừ	-	$7-2=5$
Nhân	*	$4*3=12$
Lũy thừa	**	$2**3=8$
Chia	/	$7/2=3.5$
Chia lấy thương	//	$7//2=3$
Chia lấy dư	%	$7\%2=1$

```
x=int(input("Nhập giá trị x: "))
y=int(input("Nhập giá trị y: "))
print("Tổng %d và %d bằng" % (x,y),x+y)
type(x)
```

Nhập giá trị x: 4
Nhập giá trị y: 3
Tổng 4 và 3 bằng 7

int

```
x=2-5j
y=complex(4,7)
print(x,y,x+y,x-y,x*y)
print("Kiểu biến x: ",type(x))
print("Kiểu biến y: ",type(y))
```

(2-5j) (4+7j) (6+2j) (-2-12j) (43-6j)
Kiểu biến x: <class 'complex'>
Kiểu biến y: <class 'complex'>

- Hàm **tạo số ngẫu nhiên**: Python cung cấp module random cho phép sinh số ngẫu nhiên.
 - Hàm **randint**(a,b) tạo ngẫu nhiên 1 số nguyên thuộc [a,b].
 - Hàm **uniform**(a,b) tạo ngẫu nhiên 1 số thực thuộc [a,b].

```
from random import randint, uniform
x=randint(0,45)
y=uniform(0,1)
print("Số nguyên tạo được là: ",x)
print("Số thực tạo được là: ",y)
```

Số nguyên tạo được là: 13

Số thực tạo được là: 0.749678621271894

- Một số hàm toán học thông dụng:

- Để sử dụng các hàm toán học, ta dùng câu lệnh:

`from math import *`

Tên hàm	Thực hiện
$\sin(x)$	Tính sin của góc có số đo x (radian)
$\cos(x)$	Tính cosin của góc có số đo x (radian)
$\tan(x)$	Tính tang của góc có số đo x (radian)
$\text{sqrt}(x)$	Tính căn bậc hai của x
$\text{trunc}(x)$	Trả về phần nguyên của x
π	Trả về giá trị của số pi
$\text{round}(x,n)$	Làm tròn x đến n chữ số thập phân
$\text{abs}(x)$	Trả về giá trị tuyệt đối của x
$\text{max}(a_1, a_2, \dots, a_n)$	Trả về giá trị lớn nhất trong dãy số
$\text{min}(a_1, a_2, \dots, a_n)$	Trả về giá trị nhỏ nhất trong dãy số

- Một số cách viết rút gọn:

Biểu thức	Rút gọn
$n=n+a$	$n+=a$
$n=n-a$	$n-=a$
$n=n*a$	$n*=a$
$n=n/a$	$n/=a$
$n=n//a$	$n//=a$
$n=n**a$	$n**=a$

Thực hành:

- **Bài tập C1-1:** Viết chương trình nhập vào hai giá trị x và y . Sau đó tính giá trị các biểu thức:

$$p_1 = x + 2y, \quad p_2 = x^2 - 2xy + y^2, \quad p_3 = 3\sin(x) + 4\sin^3(x),$$

Hiển thị các giá trị p_1 , p_2 , p_3 và cho biết giá trị lớn nhất trong 3 số đó.

- **Bài tập C1-2:** Viết chương trình nhập a , b , c là ba cạnh của một tam giác. Sau đó tính diện tích của tam giác đó.

Gợi ý: sử dụng công thức Heron, $S = \sqrt{p(p-a)(p-b)(p-c)}$,

$$\text{ở đó } p = \frac{1}{2}(a + b + c)$$

- **Bài tập C1-3:** Giả sử bạn mở một tài khoản tiết kiệm kì hạn 1 tháng tại một ngân hàng với lãi suất p mỗi năm. Lãi suất sẽ được trả/cộng vào tài khoản khi đáo hạn. Viết chương trình nhập vào số tiền gửi, lãi suất. Chương trình sẽ tính và hiển thị số tiền trong tài khoản sau 1, 2, và 3 tháng.

5. KIỂU DỮ LIỆU CHUỖI

- Chuỗi là một dãy các kí tự. Mọi kí tự nằm trong “ ” (*ngoặc kép*) và ‘ ’ (*ngoặc đơn*) đều là chuỗi trong Python.
- Trong Python, các kí tự trong chuỗi thuộc bảng mã Unicode hoặc bảng mã ASCII. Do đó chuỗi có hỗ trợ tiếng Việt.
- Số kí tự trong chuỗi được gọi là độ dài của chuỗi.

- Kí tự trong chuỗi được đánh chỉ số bắt đầu từ 0, 1, ... (từ trái qua phải) và bắt đầu từ -1, -2, ... (từ phải qua trái)

	H	e	l	l	o		W	o	r	l	d
Chỉ số	0	1	2	3	4	5	6	7	8	9	10
	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Một số thao tác với chuỗi:

- *Hàm lấy độ dài:* **len(<biến chuỗi>)**
- *Truy cập đến từng kí tự trong chuỗi:*

<biến chuỗi>[chỉ số]

- *Ghép chuỗi:*

<Chuỗi 1> + <Chuỗi 2> + ... + <Chuỗi n>

- *Lặp chuỗi:* **<biến chuỗi>*<số lần ghép>** hoặc

<số lần ghép>*<biến chuỗi>

- *So sánh chuỗi*: lần lượt so sánh các kí tự của các chuỗi từ trái qua phải. Nếu gặp một cặp kí tự không bằng nhau thì chuỗi có kí tự lớn hơn là chuỗi lớn hơn. Nếu một chuỗi là phần đầu của chuỗi kia thì chuỗi đó là nhỏ hơn.

```
str1, str2, str3="Tôi", "học", 'Python'

# Hiển thị 3 chuỗi
print(str1,str2,str3,sep=", ")

# Ghép chuỗi
str4=str1+" "+str2+" "+str3+"."
print("Chuỗi ghép: ",str4)
print("Độ dài = ",len(str4))
|
# Truy cập kí tự trong chuỗi
print(str4[0],str4[9],str4[8],end=" ")
print(str4[-4],str4[-3],str4[-2])

# Lặp chuỗi
str5=2*str4
print("Chuỗi lặp: ",str5)

# So sánh chuỗi
print(str1<str2, str1<str3, str1<str4)
```

```
Tôi, học, Python
Chuỗi ghép: Tôi học Python.
Độ dài = 15
T y P h o c
Chuỗi lặp: Tôi học Python.Tôi học Python.
True False True
```

■ Lấy chuỗi con:

<biến chuỗi>[a:b]

cho kết quả là chuỗi gồm các kí tự từ vị trí a đến b-1.

- ❖ **<biến chuỗi>[:a]** = Lấy a kí tự đầu của chuỗi.
- ❖ **<biến chuỗi>[a:]** = Lấy các kí tự từ chỉ số a đến cuối.
- ❖ **<biến chuỗi>[a:b:c]** = Lấy các kí tự trong chuỗi có chỉ số a, a+c, ..., a+tc < b.

■ Đảo ngược chuỗi:

<biến chuỗi>[::-1]

- Câu lệnh: **str(<biến>)** biến dữ liệu kiểu bất kì thành dữ liệu chuỗi.

L	â	p		t	r	ì	n	h		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

```

st="Lập trình Python"
print(st)
print(st[0:11])
print(st[0:11:2])
print(st[::-1])
print(st+" "+str(2020))

```

```

Lập trình Python
Lập trình P
LptìhP
nohtyP hñirt pậL
Lập trình Python 2020

```

Thực hành:

- **Bài tập C1-4:** Viết chương trình yêu cầu người dùng nhập vào lần lượt họ, đệm, và tên của mình. Chương trình sẽ in ra thông báo “Xin chào” + họ + đệm + tên người dùng được nhập. Tính và in ra độ dài họ tên người dùng.
- **Bài tập C1-5:** Viết chương trình yêu cầu người dùng nhập vào lần lượt họ, và tên của mình. Chương trình sẽ sinh ra một địa chỉ email có dạng: họ.tên.xyz@utc.edu.vn, trong đó xyz là số tự nhiên có 3 chữ số được sinh ngẫu nhiên.
- **Bài tập C1-6:** Viết chương trình nhập vào chuỗi bất kì, gồm kí tự chữ và số xen kẽ nhau. Chương trình sẽ tách chuỗi thành 2 chuỗi con, mỗi chuỗi chỉ chứa kí tự chữ hoặc số.