



# PERL & PYTHON

**Course Code: CS466**

**No. of Credit Hours: 2 credits (LEC)**

Lecturer: Dang, Tran Huu Minh

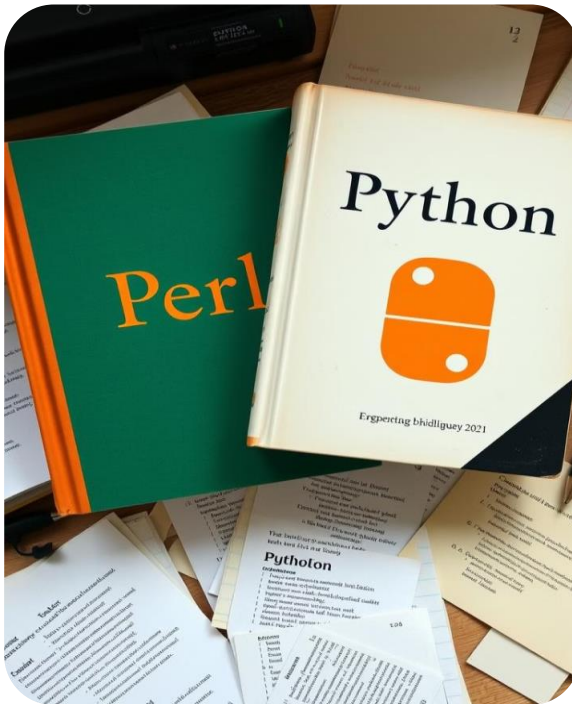
Mobile/ Zalo: 0918.763.367

Email: [tranhminhdang@dtu.edu.vn](mailto:tranhminhdang@dtu.edu.vn)



## Chapter 1. Rerview of Perl and Python

### CONTENT



Overview of the Perl programming language



Overview of the Python programming language



## GIỚI THIỆU THÔNG TIN SLIDE

**Chủ đề:**

### **TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH PYTHON**

**Thời lượng: 120 phút**

**Lecturer: Dang, Tran Huu Minh**

**Mobile/ Zalo: 0918.763.367**

**Email: [tranhminhdang@dtu.edu.vn](mailto:tranhminhdang@dtu.edu.vn)**



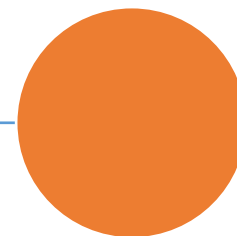


## TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH PYTHON

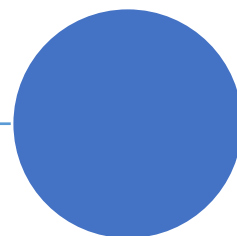
### ❑ MỤC TIÊU:

Sau khi học xong bài này, sinh viên sẽ có được những khả năng sau:

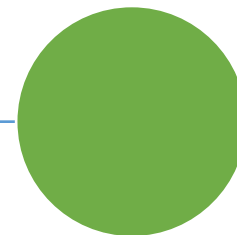
- **Hiểu và sử dụng biến, kiểu dữ liệu và các toán tử trong Python**
- **Vận dụng để viết một chương trình đơn giản bằng Python.**



Biết về Python



Cài đặt được Python



Biết các cú pháp trong Python



# TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH PYTHON

## NỘI DUNG

- **Lịch sử và Phát triển của Python**
- **Đặc điểm và Tính năng Chính của Python**
- **Cấu trúc và Cú pháp Cơ bản**
- **Ứng dụng của Perl trong Thực Tế**

## Tài Nguyên Tham Khảo

Eric Chou (2023). Mastering Python Networking, Fourth Edition.  
Packt Publishing.



## Tổng Quan về Ngôn ngữ Lập trình Python



Python là một ngôn ngữ lập trình bậc cao, đa năng và phổ biến được sử dụng trong nhiều lĩnh vực, từ phát triển web và khoa học dữ liệu đến tự động hóa và trí tuệ nhân tạo.

Python nổi tiếng với cú pháp rõ ràng, dễ đọc và dễ học, làm cho nó trở thành một lựa chọn tuyệt vời cho người mới bắt đầu và các lập trình viên có kinh nghiệm.

Sự phổ biến rộng rãi của Python là kết quả của khả năng ứng dụng đa dạng, cộng đồng hỗ trợ mạnh mẽ và các thư viện phong phú.





## Lịch sử và sự phát triển của Python



1

**1991**

Guido van Rossum tạo ra Python tại Viện nghiên cứu quốc gia về toán học và khoa học máy tính ở Hà Lan.

2

**2000**

Phiên bản Python 2.0 được phát hành với các tính năng mới như bộ thu gom rác và hỗ trợ Unicode.

3

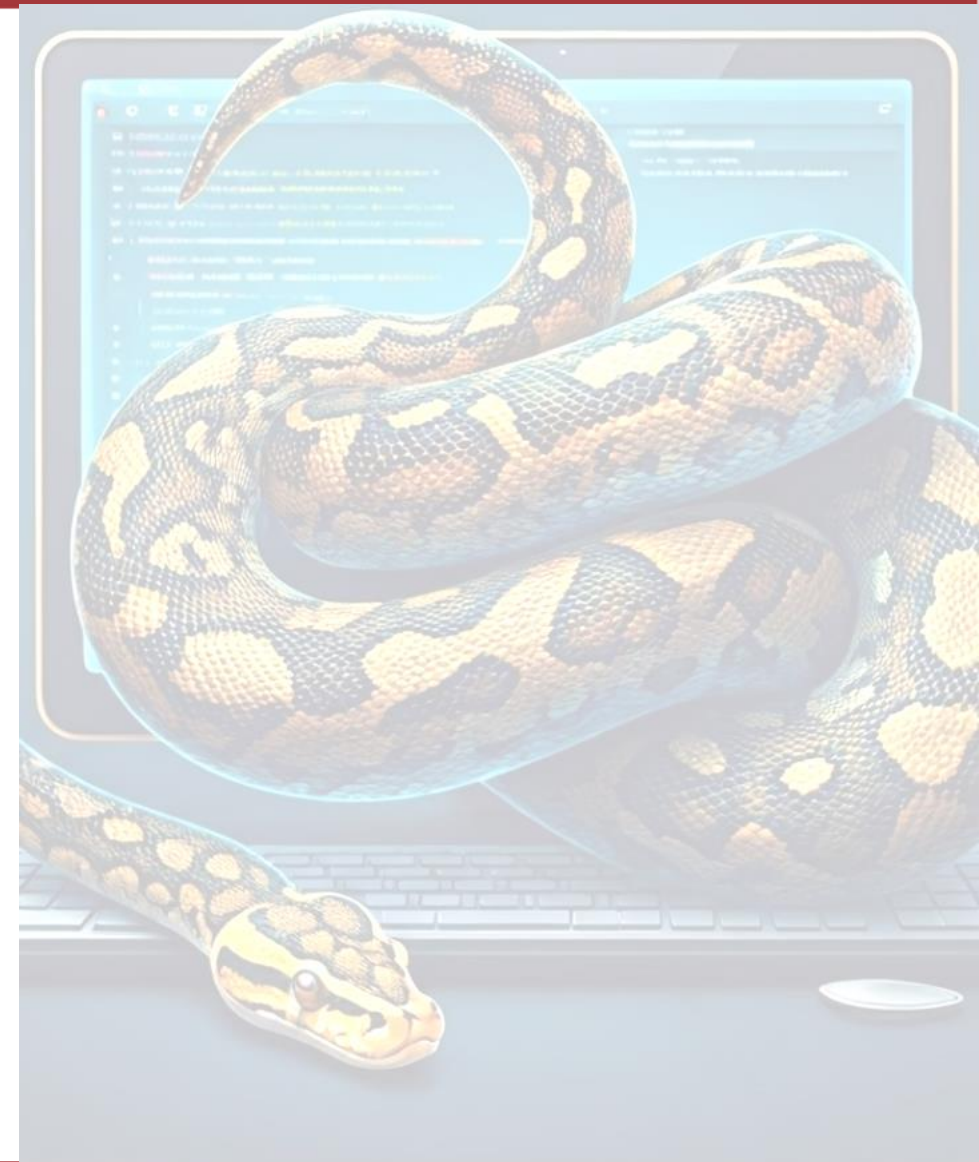
**2010**

Python 3.0 được phát hành, với các thay đổi lớn về cú pháp và khả năng tương thích ngược với các phiên bản trước.



## Đặc điểm và ưu điểm của Python

- 1 Dễ học và dễ sử dụng**  
Cú pháp đơn giản, dễ đọc và viết, thích hợp cho người mới bắt đầu.
- 2 Mạnh mẽ và linh hoạt**  
Hỗ trợ nhiều mô hình lập trình, có thể sử dụng để phát triển các ứng dụng phức tạp.
- 3 Cộng đồng lớn và hỗ trợ tốt**  
Có nhiều tài liệu, thư viện và framework hỗ trợ, cộng đồng người dùng đông đảo.
- 4 Miễn phí và mã nguồn mở**  
Có thể sử dụng và sửa đổi mã nguồn một cách tự do.







## Các lĩnh vực ứng dụng của Python

### Phát triển web

Django, Flask, Pyramid là các framework phổ biến cho phát triển web với Python.

### Khoa học dữ liệu

NumPy, Pandas, Scikit-learn là các thư viện mạnh mẽ để phân tích dữ liệu, xây dựng mô hình dự đoán.

### Trí tuệ nhân tạo

TensorFlow, PyTorch, Keras là các thư viện hỗ trợ học sâu, xử lý ngôn ngữ tự nhiên.



## Cài đặt và cấu hình Python

1

### Tải xuống Python

Truy cập trang web chính thức của Python và tải xuống xuống phiên bản phù hợp với hệ điều hành của bạn.

2

### Cài đặt Python

Chạy file cài đặt và làm theo hướng dẫn trên màn hình để cài đặt Python.

3

### Kiểm tra cài đặt

Mở terminal hoặc command prompt và gõ "python --version" để kiểm tra phiên bản Python đã được cài đặt.

4

### Cấu hình môi trường

Thiết lập các biến môi trường cần thiết để sử dụng Python một cách hiệu quả.



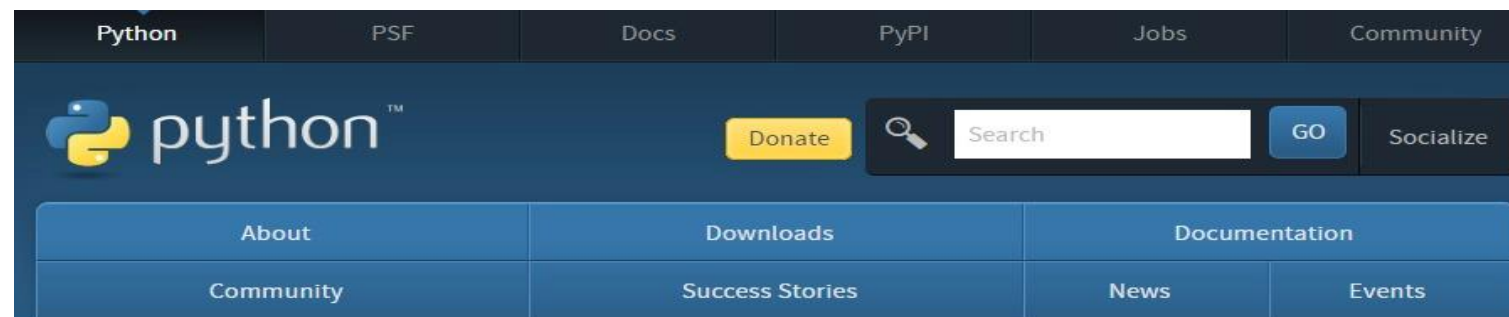


## Tải xuống trình cài đặt Python

Bước 1: Đi tới trang tải xuống Python chính thức cho Windows.

Bước 2: Tìm bản phát hành Python 3 ổn định. Hướng dẫn này đã được thử nghiệm với Python phiên bản 3.10.10.

Bước 3: Nhấp vào liên kết thích hợp cho hệ thống của bạn để tải xuống tệp thực thi: **Trình cài đặt Windows (64-bit)** hoặc **Trình cài đặt Windows (32-bit)**.



Python >>> Downloads >>> Windows

### Python Releases for Windows

- [Latest Python 3 Release - Python 3.11.2](#)

#### Stable Releases

- [Python 3.10.10 - Feb. 8, 2023](#)

**Note that Python 3.10.10 cannot be used on Windows 7 or earlier.**

- [Download Windows embeddable package \(32-bit\)](#)
- [Download Windows embeddable package \(64-bit\)](#)
- [Download Windows help file](#)
- [Download Windows installer \(32-bit\)](#)
- [Download Windows installer \(64-bit\)](#)

#### Pre-releases

- [Python 3.12.0a5 - Feb. 7, 2023](#)

- [Download Windows embeddable package \(32-bit\)](#)
- [Download Windows embeddable package \(64-bit\)](#)
- [Download Windows embeddable package \(ARM64\)](#)
- [Download Windows installer \(32-bit\)](#)
- [Download Windows installer \(64-bit\)](#)
- [Download Windows installer \(ARM64\)](#)

- [Python 3.12.0a4 - Jan. 10, 2023](#)

- [Download Windows embeddable package \(32-bit\)](#)



## Cài đặt

Bước 1: Sau khi tải xuống trình cài đặt, chẳng hạn như python-3.10.10-amd64.exe để chạy trình cài đặt Python.

Bước 2: Chọn hộp kiểm Cài đặt trình khởi chạy cho tất cả người dùng , hộp kiểm này cho phép tất cả người dùng máy tính truy cập ứng dụng trình khởi chạy Python.

Bước 3: Chọn hộp kiểm Thêm python.exe vào PATH , cho phép người dùng khởi chạy Python từ dòng lệnh.

Bước 4: Bấm chọn Install Now







Bước 5: Các tính năng tùy chọn bao gồm các công cụ và tài nguyên phổ biến dành cho Python và bạn có thể cài đặt tất cả chúng, ngay cả khi bạn không định sử dụng chúng.

Bước 6: Bấm Next





Python 3.10.10 (64-bit) Setup

## Advanced Options

- ☐ Install Python 3.10 for all users
- ☒ Associate files with Python (requires the 'py' launcher)
- ☒ Create shortcuts for installed applications
- ☒ Add Python to environment variables
- ☐ Precompile standard library
- ☐ Download debugging symbols
- ☐ Download debug binaries (requires VS 2017 or later)

Customize install location

C:\Users\4andi\AppData\Local\Programs\Python\Python310

Browse

Back



Cancel

## Bước 7: Hộp thoại Tùy chọn nâng cao hiển thị.

*Chọn các tùy chọn phù hợp với yêu cầu của bạn:*

- **Cài đặt cho tất cả người dùng** : được khuyến nghị nếu bạn không phải là người dùng duy nhất trên máy tính này
- **Liên kết tệp với Python** : được khuyến nghị, vì tùy chọn này liên kết tất cả các loại tệp Python với trình khởi chạy hoặc trình chỉnh sửa
- **Tạo lối tắt cho các ứng dụng đã cài đặt** : khuyến nghị bật lối tắt cho ứng dụng Python
- **Thêm Python vào các biến môi trường** : được khuyến nghị để kích hoạt khởi chạy Python
- **Thư viện chuẩn biên dịch trước** : không bắt buộc, nó có thể làm hỏng quá trình cài đặt
- **Tải xuống các biểu tượng gỡ lỗi và Tải xuống các tệp nhị phân gỡ lỗi** : chỉ được khuyến nghị nếu bạn dự định tạo tiện ích mở rộng C hoặc C++

## Bước 8: Bấm vào Cài đặt (Install) để bắt đầu cài đặt.





Bước 9: Sau khi quá trình cài đặt hoàn tất, thông báo **Thiết lập thành công** sẽ hiển thị.

Python 3.10.10 (64-bit) Setup



## Setup was successful

New to Python? Start with the [online tutorial](#) and [documentation](#). At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See [what's new](#) in this release, or find more info about [using Python on Windows](#).



### Disable path length limit

Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX\_PATH" limitation.

Close



## Phần mềm soạn thảo

### Visual Studio Code

Visual Studio Code là một trình soạn thảo mã phổ biến, được hỗ trợ bởi Microsoft, cung cấp nhiều tính năng hữu ích cho lập trình viên Python, bao gồm tự động hoàn thành mã, gỡ lỗi và hỗ trợ kiểm tra mã.

### PyCharm

PyCharm là một IDE chuyên nghiệp dành cho Python, cung cấp đầy đủ các tính năng hỗ trợ phát triển Python, bao gồm gỡ lỗi nâng cao, tự động hoàn thành mã, kiểm tra mã và tích hợp với các công cụ kiểm soát phiên bản.

### Sublime Text

Sublime Text là một trình soạn thảo mã nhẹ và nhanh, có thể được sử dụng để phát triển Python. Nó cung cấp nhiều tính năng hữu ích, bao gồm tự động hoàn thành mã, gỡ lỗi và hỗ trợ nhiều ngôn ngữ.

### Atom

Atom là một trình soạn thảo mã mã nguồn mở được phát triển bởi GitHub, cung cấp khả năng tùy chỉnh cao và nhiều gói mở rộng cho lập trình Python.



## Cú pháp cơ bản của Python

### Biến

Khai báo biến bằng cách sử dụng tên biến và gán giá trị cho nó.

### Kiểu dữ liệu

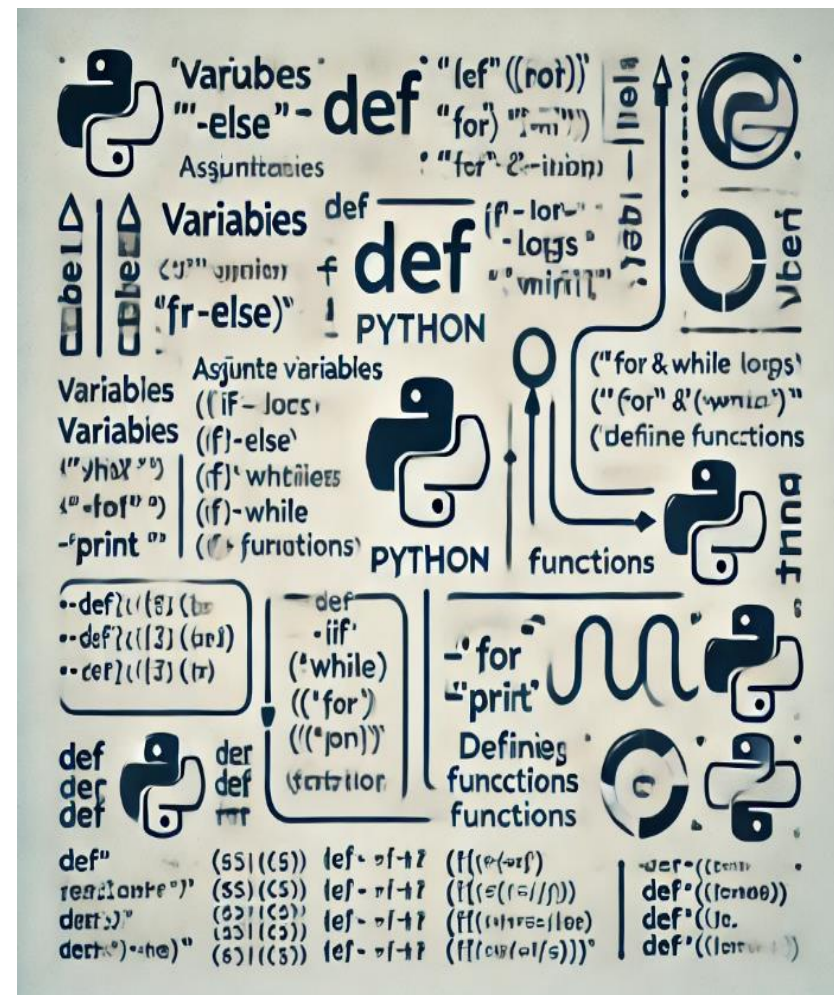
Python hỗ trợ nhiều kiểu dữ liệu như số nguyên, số thực, chuỗi, danh sách, tuple, từ điển.

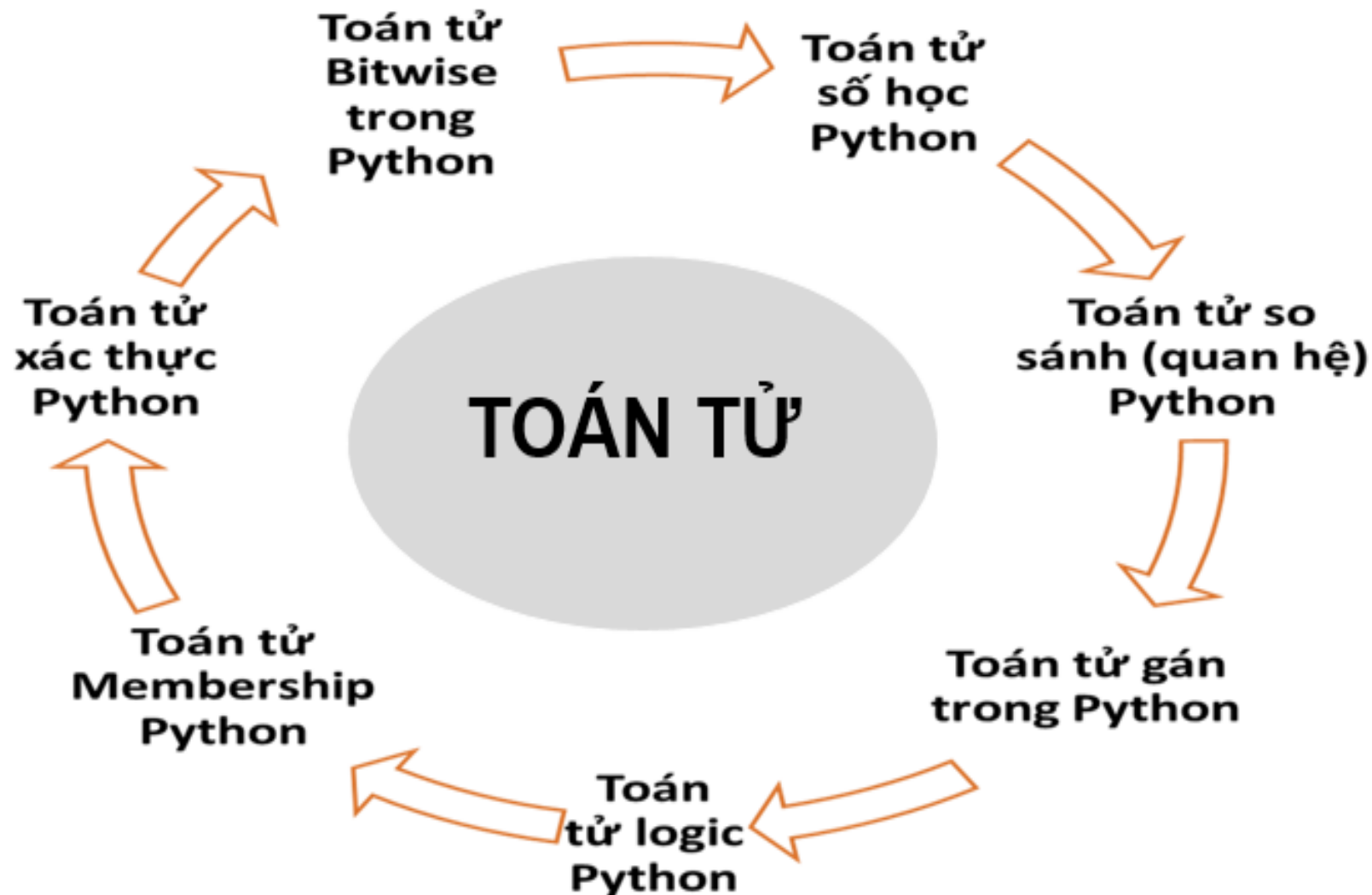
### Toán tử

Sử dụng các toán tử cộng, trừ, nhân, chia, modulo, lũy thừa, so sánh, logic.

### Câu điều khiển

Sử dụng các câu lệnh if-else, for, while, break, continue để kiểm soát luồng thực thi.









## Toán Tử Số Học Trong Python

Operators	Name
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus(Chia lấy dư)
**	Exponentiation(Lũ y thừa)
//	Floor Division(Chia làm tròn lấy phần nguyên)

Toán tử	Cách viết	Ví dụ	Ý nghĩa
Cộng	$x + y$	$20 + 3 = 23$	Toán tử cộng các giá trị lại với nhau
Trừ	$x - y$	$20 - 3 = 17$	Toán tử trừ các giá trị lại với nhau
Nhân	$x * y$	$20 * 3 = 60$	Toán tử nhân các giá trị lại với nhau
Chia	$x / y$	$20 / 3 = 6.666$	Toán tử chia các giá trị cho nhau
Lấy thương	$x // y$	$20 // 3 = 6$	Toán tử chia lấy phần thương được làm tròn xuống
Lấy dư	$x \% y$	$20 \% 3 = 2$	Toán tử chia lấy phần dư
Đổi dấu	$-x$	$-20 = 20$	Toán tử đổi dấu của số
Giữ dấu	$+x$	$+20 = 20$	Toán tử giữ nguyên dấu của số
Mũ	$x ** y$	$20^3 = 8000$	Toán tử mũ xy



## Toán tử so sánh (quan hệ) Python

Toán tử	Tên	Ví dụ minh họa
==	So sánh bằng	$X == Y$
<	So sánh nhỏ hơn	$X < Y$
>	So sánh lớn hơn	$X > Y$
<=	So sánh nhỏ hơn hoặc bằng	$X \leq Y$
>=	So sánh lớn hơn hoặc bằng	$X \geq Y$
!=	So sánh không tương đương	$X \neq Y$





## Toán tử gán trong Python

Toán tử	Toán tử	Ví dụ	Tương đương
=	=	<code>x = 4</code>	<code>x = 4</code>
+=	+=	<code>x += 4</code>	<code>x = x + 4</code>
-=	-=	<code>x -= 4</code>	<code>x = x - 4</code>
*=	*=	<code>x *= 4</code>	<code>x = x * 4</code>
/=	/=	<code>x /= 4</code>	<code>x = x / 4</code>
%=	%=	<code>x %= 4</code>	<code>x = x % 4</code>
//=	//=	<code>x //= 4</code>	<code>x = x // 4</code>
**=	**=	<code>x **= 4</code>	<code>x = x ** 4</code>



## Toán tử logic Python

Toán tử	Chú thích	Cú pháp
and	Return <b>True</b> Nếu cả <b>hai</b> biểu thức <b>đúng</b>	<b>x and y</b>
or	Return <b>True</b> Nếu <b>một</b> trong các biểu thức là <b>đúng</b>	<b>x or y</b>
not	<b>Phủ định</b> kết quả Trả về <b>True</b> nếu kết quả là <b>False</b> và ngược lại	<b>not x</b>



## Toán tử Membership Python

Toán tử	Tên gọi	Chú thích
&	AND	Nếu cả 2 bit là 1, giá trị trả về là 1, ngược lại trả về 0
	OR	Nếu một trong hai bit là 1, giá trị trả về là 1, ngược lại trả về 0.
^	XOR	Nếu hai <b>bit</b> khác nhau, giá trị trả về là 1, ngược lại trả về 0.
~	NOT	Đảo ngược tất cả các <b>bit</b> , 0 thành 1 và 1 thành 0.
<<	Right shift	Dịch chuyển tất cả các <b>bit</b> sang bên trái.
>>	Left shift	Dịch chuyển tất cả các <b>bit</b> sang bên phải ngoại trừ <b>bit</b> đầu tiên



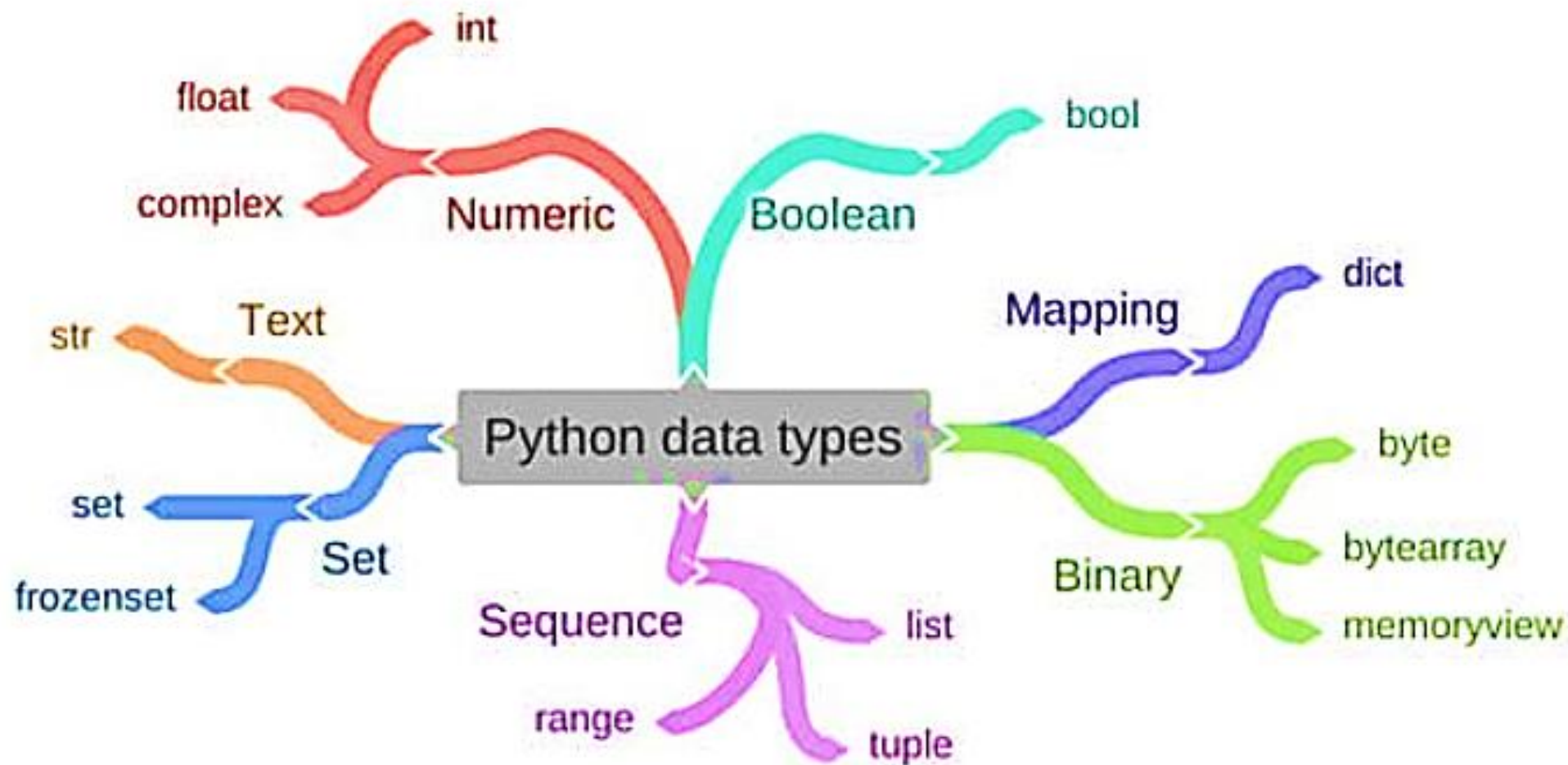
## Toán tử xác thực Python

Toán Tử	Chú Thích	Ví Dụ X=5 Y=[1,3,8,10]
in	Nếu 1 đối số thuộc một tập đối số nó sẽ trả về True và ngược lại/	X in Y → False
not in	Nếu 1 đối số không thuộc một tập đối số nó sẽ trả về True và ngược lại/	X not in Y → True
is	Trả về True nếu 2 toán hạng giống nhau	X is Y → False
is not	Trả về True nếu hai toán hạng khác nhau	X is not Y → True



## Toán tử Bitwise trong Python

Toán Tử	Ví Dụ $a = 12 \rightarrow$ và $b = 15$
&	$(a \& b) = 12$ (00001100)
	$(a   b) = 14$ (00001111)
^	$(a \wedge b) = 2$ (00000010)
~	$(-a) = -13$ (00001101)
<<	$a \ll a = 49152$
>>	$a \gg a = 0$







## Kiểu dữ liệu chuỗi str

Kiểu dữ liệu chuỗi (str) được sử dụng để lưu trữ văn bản.

### 1 Khả năng kết hợp

Kết hợp hai chuỗi bằng toán tử "+"

### 2 Chuyển đổi

Chuyển đổi chuỗi sang các kiểu dữ liệu khác, chẳng hạn như số.

### 3 Xóa bỏ khoảng trắng

Loại bỏ khoảng trắng ở đầu và cuối chuỗi.

### 4 Tìm kiếm

Tìm kiếm một chuỗi con trong chuỗi lớn hơn.

Ví dụ, "Hello world!", "123", "abc" là các chuỗi. Các chuỗi có thể được kết hợp, cắt, thay thế và xử lý theo nhiều cách khác nhau.

### "data types"

- **string:** "321" # character(s)
- **integer:** 321 # whole number
- **float:** 321.0 # decimal
- **Boolean:** True or False # binary
- **array:** ["321", 321, 321.0, True]  
# a list of values





## Kiểu số nguyên int

Kiểu dữ liệu số nguyên (int) được sử dụng để lưu trữ số nguyên, bao gồm cả số dương, số âm và số 0.

Ví dụ, 10, -5, 0 là các số nguyên.



Toán tử	Tác dụng
+	Cộng
-	Trừ
*	Nhân
/	Chia
//	Chia lấy phần nguyên
%	Chia lấy phần dư
	Lũy thừa



## Kiểu dữ liệu số thực float

Kiểu dữ liệu số thực (float) được sử dụng để lưu trữ các số có phần thập phân, chẳng hạn như 3.14, -2.5, 0.0.

### Chuyển đổi

Chuyển đổi số thực thành số nguyên hoặc chuỗi.

### Làm tròn

Làm tròn số thực lên hoặc xuống số nguyên gần nhất.

### So sánh

So sánh hai số thực bằng các toán tử so sánh.



Kiểu dữ liệu loại trình tự lưu trữ một tập hợp các phần tử theo thứ tự. List, tuple và range là ba loại trình tự phổ biến trong lập trình.

## 1 List

List là kiểu dữ liệu có thể thay đổi, cho phép bạn thêm, xóa hoặc sửa đổi các phần tử của nó.

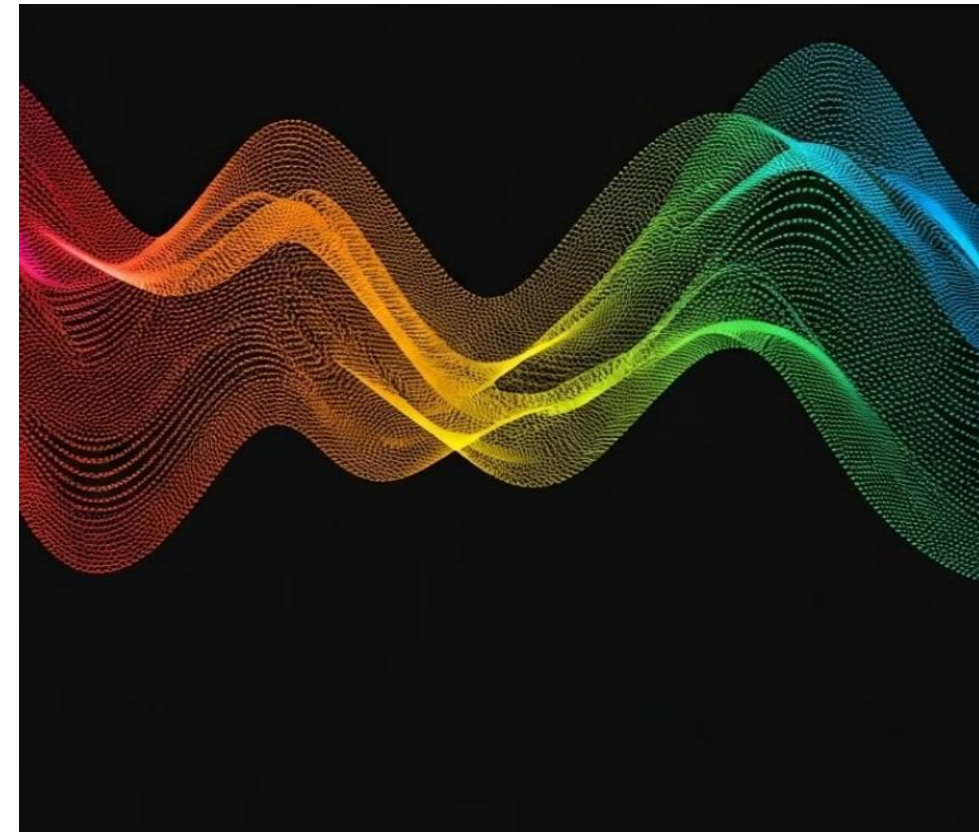
## 2 Tuple

Tuple là kiểu dữ liệu không thể thay đổi, nghĩa là các phần tử của nó không thể được sửa đổi sau khi tạo.

## 3 Range

Range tạo ra một chuỗi số liên tiếp. Range thường được sử dụng trong các vòng lặp.

## Kiểu dữ liệu loại trình tự: **list, tuple, range**





## Kiểu dữ liệu ánh xạ: dict

Kiểu dữ liệu ánh xạ (dict) được sử dụng để lưu trữ dữ liệu theo cặp khóa-giá trị. Mỗi khóa phải duy nhất và được sử dụng để truy cập giá trị tương ứng.

1

### Tạo từ điển

Tạo một từ điển mới với các cặp khóa - giá trị.

2

### Thêm cặp khóa - giá trị

Thêm một cặp khóa-giá trị mới vào từ điển.

3

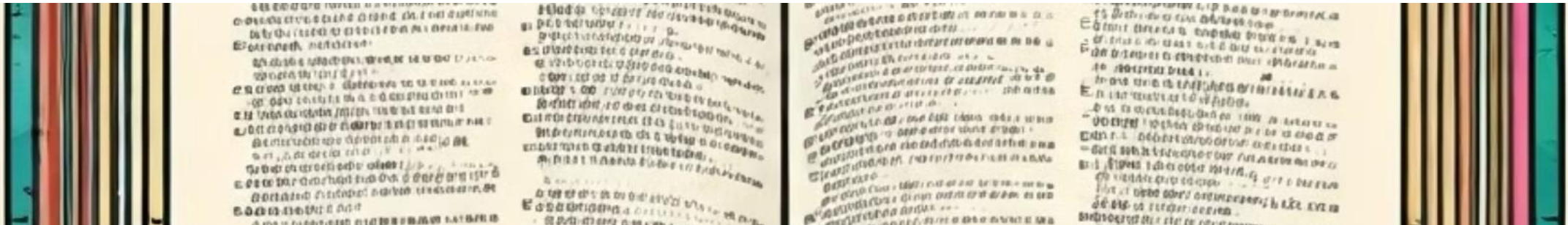
### Truy cập giá trị

Truy cập giá trị tương ứng với một khóa đã biết.

4

### Sửa đổi giá trị

Sửa đổi giá trị tương ứng với một khóa.







Kiểu dữ liệu set được sử dụng để lưu trữ một tập hợp các phần tử duy nhất, không có thứ tự. Frozenset là một kiểu dữ liệu tương tự, nhưng không thể thay đổi.

## Kiểm tra phần tử

Kiểm tra xem một phần tử có tồn tại trong set hay không.

## Thao tác tập hợp

Thực hiện các phép toán tập hợp như hợp, giao, hiệu.

## Loại bỏ trùng lặp

Loại bỏ các phần tử trùng lặp trong một danh sách.

## Kiểu dữ liệu: Set, set frozenset







## Kiểu boolean: bool

Kiểu dữ liệu boolean (bool) đại diện cho giá trị đúng hoặc sai. True và False là hai giá trị boolean duy nhất.

### ✓ Kết quả so sánh

Kết quả của các phép toán so sánh thường là boolean.

### ? Điều kiện

Các điều kiện trong câu lệnh điều khiển thường sử dụng giá trị boolean.

### | Logic

Các toán tử logic (AND, OR, NOT) hoạt động trên các giá trị boolean.





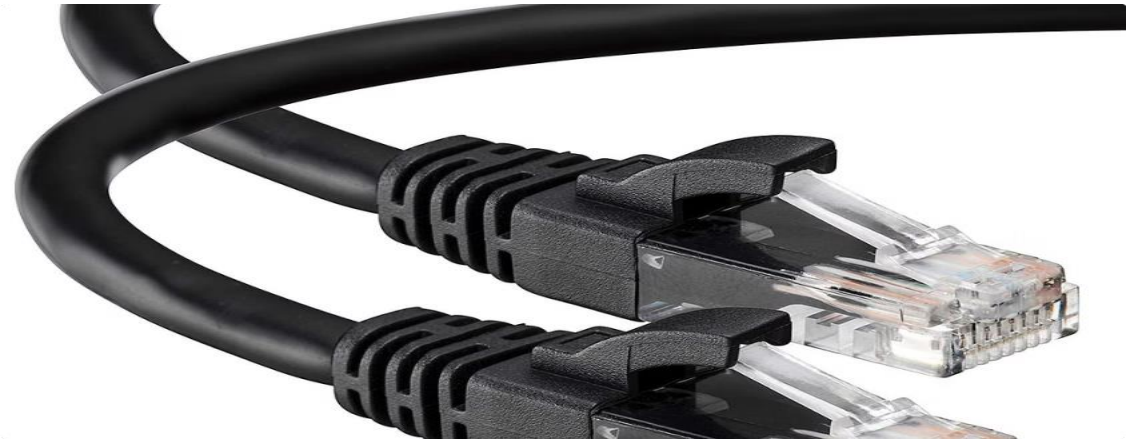
## Kiểu nhị phân: bytes, bytearray, memoryview

Kiểu dữ liệu nhị phân được sử dụng để lưu trữ dữ liệu ở dạng nhị phân, bao gồm các byte. Bytes, bytearray và memoryview là các kiểu dữ liệu nhị phân phổ biến.



### Lưu trữ dữ liệu

Các kiểu dữ liệu nhị phân được sử dụng để lưu trữ dữ liệu trên đĩa cứng, ổ đĩa flash và các thiết bị lưu trữ khác.



### Truyền dữ liệu

Dữ liệu được truyền qua mạng thường được mã hóa thành dạng nhị phân.



## Đơn/ Ghép

### Ví dụ

```
# Câu lệnh đơn
Print("Câu lệnh đơn")
# Câu lệnh ghép
Clg="Câu lệnh ghép"
Print(clg)
```

01

02

## FOR

### Ví dụ

```
chuoai = ['a','b','c']
for i in range(len(chuoai)):
    print ("chu ",chuoai[i])
```

03

04

## break, continue, pass

### Ví dụ

```
for i in range(0,15):
    if i > 6:
        print ("...",i)
        break
```

05

### Ví dụ

```
x = 10
if x > 5:
    print("x lớn hơn 5")
else:
    print("x nhỏ hơn hoặc bằng 5")
```

IF

### Ví dụ

```
count = 0
while (count < 9):
    print ('So thu tu :', count)
    count = count + 1
print ("Ket thuc")
```

WHILE





## Đơn/ Ghép

### Ví dụ

```
# Câu lệnh đơn  
Print("Câu lệnh đơn")  
# Câu lệnh ghép  
Clg="Câu lệnh ghép"  
Print(clg)
```

01

- **Câu lệnh đơn:** là những câu lệnh chỉ chiếm một dòng trong mã nguồn.

- Câu lệnh gán (assignment):  
**`x = 10`**
- Câu lệnh gọi hàm (function call):  
**`print("Hello, world!")`**
- Câu lệnh nhập liệu (input):  
**`name = input("Nhập n: ")`**
- Câu lệnh điều kiện đơn giản:  
**`if x > 5: print("x lớn hơn 5")`**



## Đơn/ Ghép

### Ví dụ

```
# Câu lệnh đơn  
Print("Câu lệnh đơn")  
# Câu lệnh ghép  
Clg="Câu lệnh ghép"  
Print(clg)
```

01

- **Câu lệnh ghép:** bao gồm một hoặc nhiều câu lệnh, và có thể chứa các câu lệnh khác lồng nhau. Chúng thường được xác định bởi các từ khóa như **if**, **for**, **while**, **def**, và **class**. Câu lệnh ghép luôn có phần thân **được thụt vào** (indented) để biểu thị các câu lệnh con.

```
count = 0  
while count < 5:  
    print(count)  
    count += 1
```






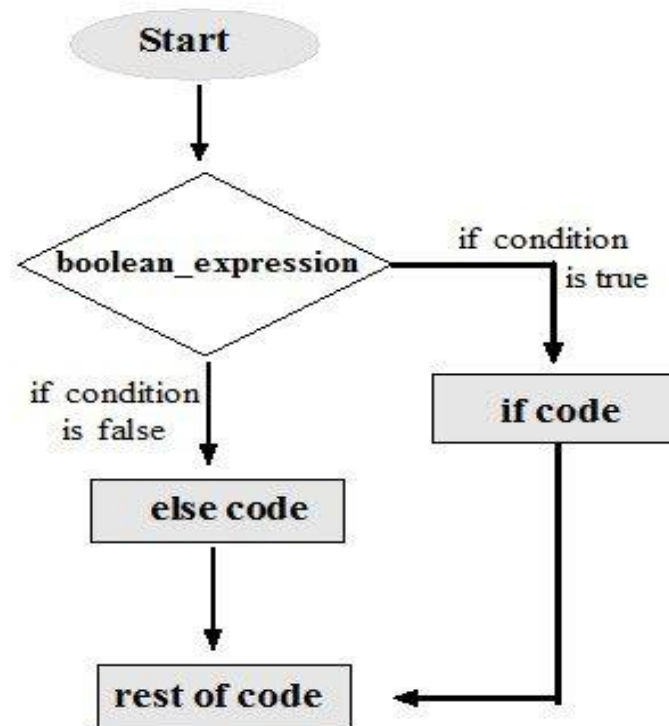


## 2.1 Câu lệnh IF trong Python


02

IF

Câu lệnh if là một trong những cấu trúc điều khiển quan trọng trong lập trình, được sử dụng để kiểm tra một điều kiện và thực hiện các khối mã khác nhau dựa trên kết quả của điều kiện đó



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



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





02

**IF**

Ví dụ

```
x = 10
if x > 5:
    print("x lớn hơn 5")
```

```
x = 10
if x >= 5:
    print("x lớn hơn và bằng 5")
else:
    print("x nhỏ hơn 5")
```

```
x = 10
y = 20
if x > 5 and y > 15:
    print("True")
```

```
x = 10
if x > 5:
    print("x lớn hơn 5")
elif x == 5:
    print("x bằng 5")
else:
    print("x nhỏ hơn ")
```

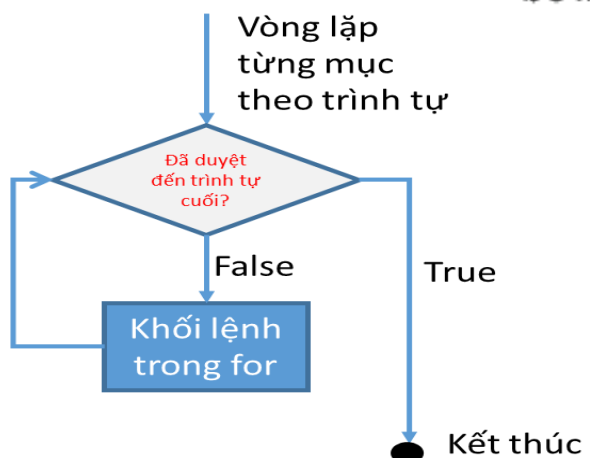
```
x = 10
y = 20
if x > 5:
    if y > 15:
        print("x lớn hơn 5 and y lớn hơn 15")
```





## FOR

03



*Var*

*It takes items from  
iterable one by one*

*Iterable*

*It's a collection of objects  
(like a list, tuple etc.)*

`for var in iterable:`

*Indentation*  
*Loop body must  
be indented*

`statement`  
`statement`  
`...`

*Loop body*

*It is executed once for  
each item in iterable*

`else:`

`statement`  
`...`

*Else clause*

*It is executed if the loop  
terminates naturally*

`following_statement`



## Ví dụ

03

## FOR

```
numbers = [1, 2, 3, 4, 5]
for num in numbers:
    print(num)
```

```
word = "Python"
for letter in word:
    print(letter)
```

```
person = {'name': 'Thu',
          'age': 25,
          'city': 'VN'}
for key in person:
    print(key, person[key])
```

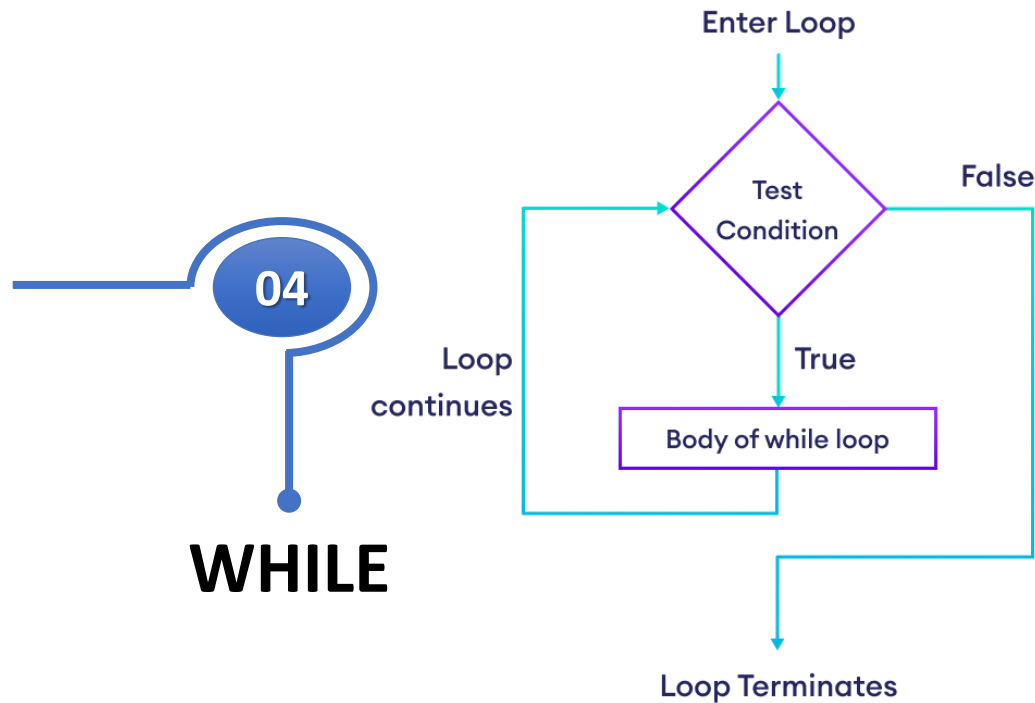
```
fruits = ['apple', 'banana', 'cherry']
for index, fruit in enumerate(fruits):
    print(index, fruit)
```

```
for i in range(5):
    print(i)
    # Kết quả: 0, 1, 2, 3, 4

for i in range(1, 6):
    print(i)
    # Kết quả: 1, 2, 3, 4, 5

for i in range(1, 10, 2):
    print(i)
    # Kết quả: 1, 3, 5, 7, 9
```





$i = 0$  ← counter

**while**  $i < 5$ : ← expression

*while Block*

$i += 1$  ← increment counter

if expression evaluates to True this block executes

if no **break** statements are encountered after **while** loop block **else** executes

**else:**

*Else Block*



code

```
1 a = 1
2 while a < 10:
3     print (a)
4     a += 2
```

output

04

**WHILE**

Ví dụ

variables





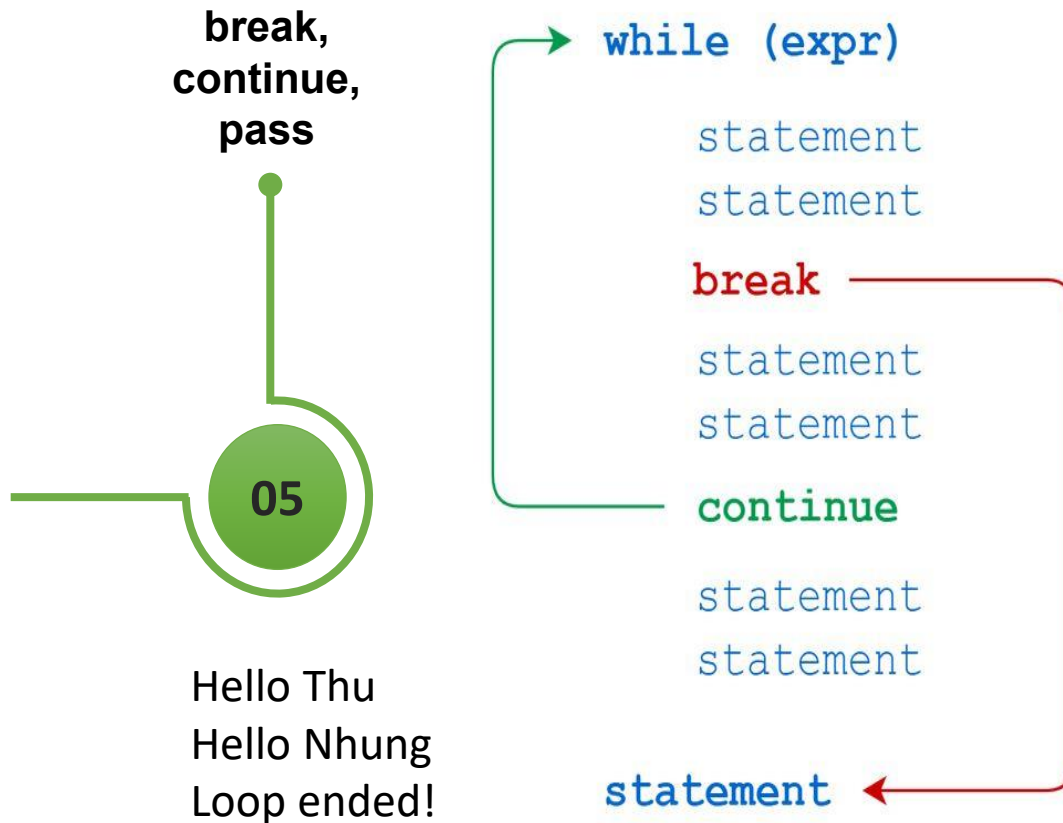
```
1 numbers = [12, 37, 5, 42, 8, 3]
2 even = []
3 odd = []
4 while len(numbers) > 0 :
5     number = numbers.pop()
6     if(number % 2 == 0):
7         even.append(number)
8     else:
9         odd.append(number)
```

---

04

## WHILE

Ví dụ



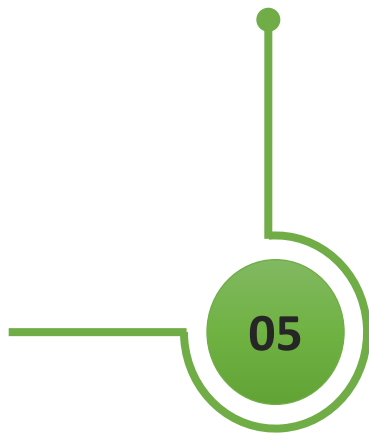
```
Ten = ['Thu', 'Khue', 'Nhung']
for n in Ten:
    if len(n) == 4:
        continue
    print('Hello', n)
    if n == 'Nhung':
        break
    if n == 'Thu':
        pass
print('Loop ended!')
```



## break

Từ khóa break được sử dụng để thoát khỏi vòng lặp sớm, ngay lập tức kết thúc vòng lặp khi được gọi.

break,  
continue,  
pass



Ví dụ

```
for i in range(10):  
    if i == 5:  
        break  
    print(i)
```

```
count = 0  
while count < 10:  
    print(count)  
    if count == 5:  
        break  
    count += 1
```



## continue

Từ khóa `continue` được sử dụng để bỏ qua phần còn lại của khối mã trong vòng lặp hiện tại và tiếp tục với lần lặp tiếp theo.

break,  
continue,  
pass

05

Ví dụ

```
for i in range(10):  
    if i % 2 == 0:  
        continue  
    print(i)
```

1  
3  
5

```
count = 0  
while count < 10:  
    count += 1  
    if count % 2 == 0:  
        continue  
    print(count)
```



## pass

Từ khóa `pass` không thực hiện hành động gì cả. Nó được sử dụng như một placeholder trong các khối mã hoặc vòng lặp khi bạn chưa biết phải viết gì ở đó.

break,  
continue,  
pass

05

Ví dụ

```
for i in range(5):  
    pass # Placeholder, không làm gì ở đây  
  
if x > 0:  
    pass # Placeholder, không làm gì ở đây
```





## Các thư viện và framework phổ biến trong Python



Thư viện	Mô tả
NumPy	Xử lý mảng đa chiều hiệu quả.
Pandas	Phân tích và xử lý dữ liệu dạng bảng.
Scikit-learn	Học máy và khai thác dữ liệu.
TensorFlow	Học sâu, trí tuệ nhân tạo.
Django	Framework phát triển web.
Flask	Framework phát triển web nhẹ.



## Ứng dụng Python trong quản trị mạng



### Cấu hình mạng

Sử dụng Python để tự động hóa quá trình cấu hình thiết bị mạng như router, switch.



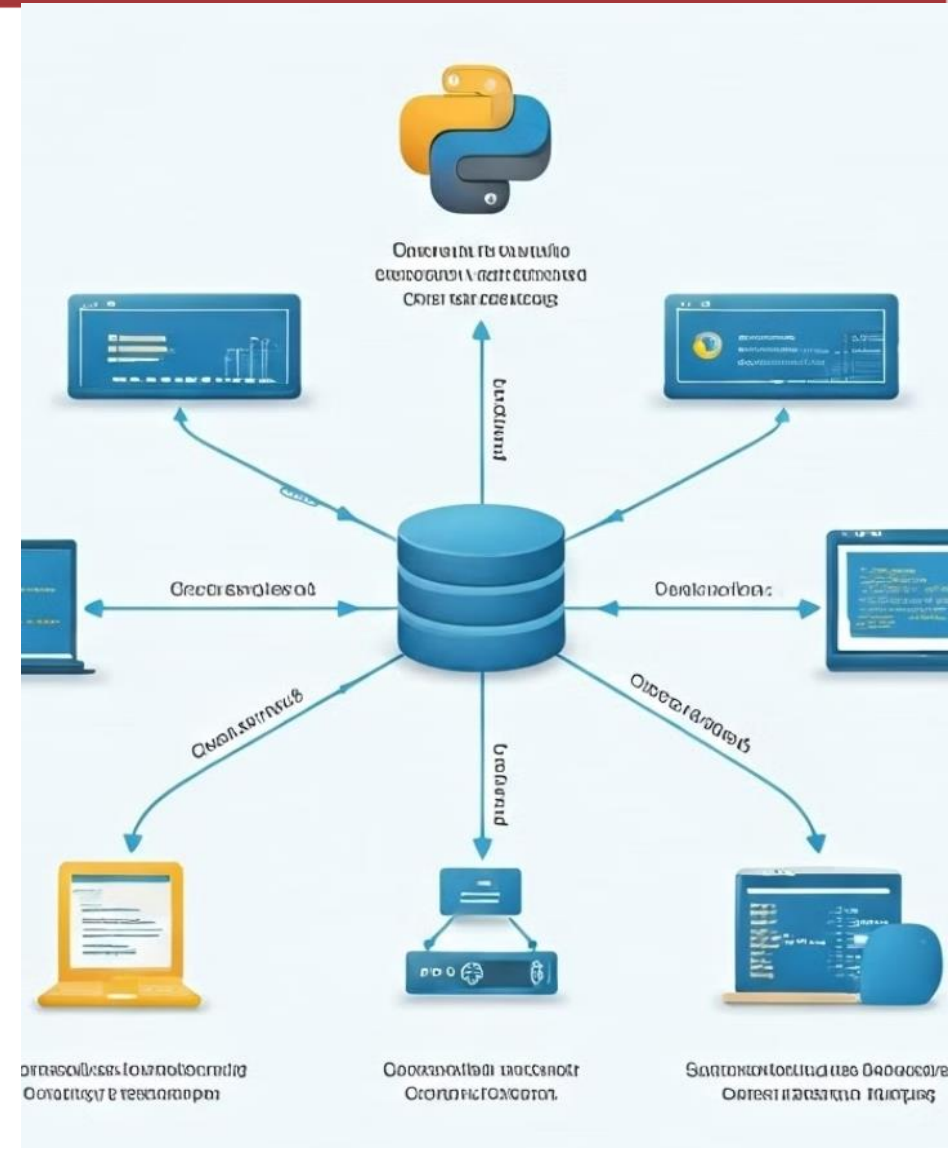
### Bảo mật mạng

Viết kịch bản để tăng cường bảo mật cho các chương trình hoạt động trong môi trường mạng.



### Quản lý mạng

Sử dụng các công cụ dựa trên Python để giám sát và quản lý hiệu suất mạng.





# Q&A

## Tổng quan về ngôn ngữ lập trình Python



## Thảo luận

1. Python có những ưu điểm gì so với các ngôn ngữ lập trình khác như Java, C++ hay JavaScript?
2. Python sử dụng trong những lĩnh vực nào và tại sao nó được ưa chuộng trong khoa học dữ liệu và học máy?
3. Python có những thư viện và framework nào hỗ trợ lập trình mạng, và chúng hoạt động như thế nào trong việc giao tiếp qua các giao thức mạng (ví dụ: TCP/IP, HTTP, FTP)?
4. Python có thể được sử dụng để tăng cường bảo mật trong lập trình mạng như thế nào?
5. Python xử lý các tác vụ đồng thời và bất đồng bộ trong môi trường mạng như thế nào?



## Q & A



Câu 1:

Python thường được sử dụng cho mục đích gì trong lĩnh vực phát triển phần mềm?

- a) Phân tích dữ liệu và khoa học dữ liệu
- b) Thiết kế đồ họa và hoạt ảnh
- c) Quản lý cơ sở dữ liệu
- d) Tất cả các phương án trên







## Q & A



Câu 2:

Python được sử dụng phổ biến trong lĩnh vực nào liên quan đến trí tuệ nhân tạo (AI)?

- a) Machine learning và deep learning
- b) Đồ họa máy tính
- c) Phát triển ứng dụng di động
- d) Quản lý dự án





## Q & A



Câu 3:

Django và Flask là gì?

- a) Ngôn ngữ lập trình Python
- b) Các loại cơ sở dữ liệu
- c) Framework phát triển web bằng Python
- d) Thư viện đồ họa





## Q & A



Câu 4:

Python có thể được sử dụng để viết ứng dụng di động cho hệ điều hành nào?

- a) Windows Phone
- b) Android và iOS
- c) Linux
- d) macOS



## Q & A



Câu 5:

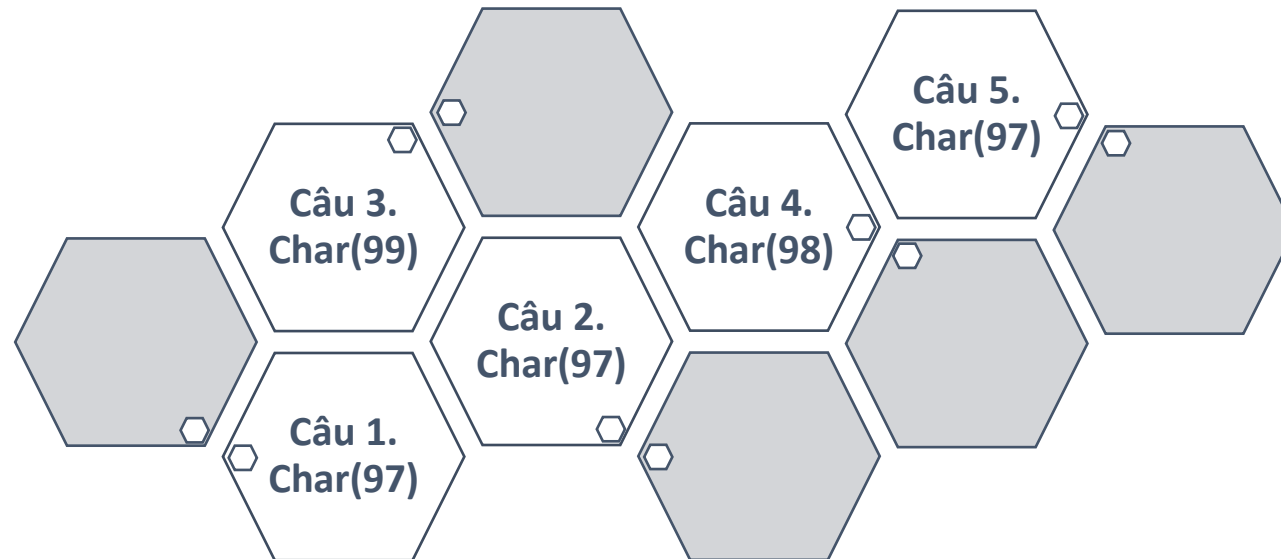
Python có khả năng tương tác với các cơ sở dữ liệu như MySQL, PostgreSQL và SQLite thông qua gì?

- a) Thư viện SQLAlchemy
- b) Thư viện Tensorflow
- c) Thư viện Matplotlib
- d) Thư viện BeautifulSoup





## Q & A







## TÀI LIỆU THAM KHẢO

### Sách & Giáo trình

- [1] Lê Văn Cường, *Python cơ bản và ứng dụng*, NXB Đại học Quốc gia Hà Nội, 2019
- [2] Randal L. Schwartz, Tom Phoenix & Brian D Foy (2008). *Learning Perl, 5<sup>th</sup> Edition*. O'Reilly Media.
- [3] Mark Lutz (2013). *Learning Python, 5<sup>th</sup> Edition*. O'Reilly Media.Stein,
- [4] Lincoln D. (2001). *Network Programming with Perl*. Addison-Wesley.
- [5] Rhodes, Brandon & Goerzen, John (2014). *Foundations of Python Network Programming, 3rd Edition*. U.S.: Apress.

### Website & Youtube

- [6] [www.python.org](http://www.python.org)
- [7] [www.perl.org](http://www.perl.org)
- [8] <https://www.youtube.com/c/TheNetNinja/playlists>
- [9] <https://www.youtube.com/watch?v=T11QYVfZoD0>
- [10] [https://www.youtube.com/watch?v=xbT7Pvh\\_5LQ](https://www.youtube.com/watch?v=xbT7Pvh_5LQ)

