1. **Phân biệt toán tử định dạng chuỗi và hàm định dạng chuỗi có sẵn trong gói thư viện chuẩn Python? Cho năm ví dụ minh họa tương ứng?**

**- Toán tử định dạng chuỗi:**

Trong Python, có nhiều cách để định dạng chuỗi, phổ biến nhất là sử dụng toán tử định dạng chuỗi (%), hàm định dạng chuỗi (str.format()), và f-string

**+ Định dạng chuỗi (%):** Bằng cách sử dụng toán tử định dạng % có thể định dạng một số hoặc chuỗi và chèn vào chuỗi ban đầu để tạo thành một chuỗi mới. Cách làm này tương đồng với hàm printf trong các ngôn ngữ lập trình khác.

**Ví dụ:**

**+ Hàm định dạng chuỗi (str.format()):** Bản thân biến dạng chuỗi là một đối tượng str có một hàm nội tại là format() dùng để định dạng nội dung.

**Ví dụ:**

ten\_nguoi\_dung = "Nguyen Van A"

cau\_chao = "Chao ban {}!"

cau\_chao\_a = cau\_chao.format(ten\_nguoi\_dung)

print(cau\_chao\_a)

**+ Hàm định dạng chuỗi f-string:** Định dạng f-string khá đơn giản khi sử dụng, tất cả những gì cần là một chữ f đứng trước một chuỗi thì tất cả các biểu thức (nằm trong cặp đóng mở ngoặc nhọn) sẽ được thực thi và chuyển đổi thành chuỗi và thay thế vào biểu thức đó.

**Ví dụ:**

age = 35

print(f"Tuổi của bạn là: {age}")

**- Hàm định dạng chuỗi có sẵn trong gói thư viện chuẩn Python:** Hàm function trong Python được hiểu là tập hợp của một khối code được viết ra, nhóm lại với nhau và đặt tên để có thể tái sử dụng cho những lần tiếp theo một cách nhanh chóng mà không cần phải viết lại toàn bộ đoạn code.Hàm giúp chia chương trình Python thành những khối, phần, mô -đun nhỏ hơn để dễ dàng tổ chức và quản lý hơn khi chương trình Python quá lớn hoặc cần được mở rộng.

**Ví dụ:**

print("Analytics Vidhya is the Largest Data Science Community over whole world")

Kết quả:

Analytics Vidhya is the Largest Data Science Community over whole world

**Ví dụ:**

Hàm type trả về kiểu của đối tượng cụ thể.

list\_of\_fruits = ('apple', 'banana', 'cherry', 'mango')

print(type(list\_of\_fruits))

Kết quả: <class 'tuple'>

1. **Viết chương trình xuất ra số ngẫu nhiên trong một đoạn bất kỳ bất cho trước?**

import random

# Nhập vào đoạn bất kỳ

min\_val = int(input("Nhập giá trị nhỏ nhất: "))

max\_val = int(input("Nhập giá trị lớn nhất: "))

# Sinh số ngẫu nhiên trong đoạn [min\_val, max\_val]

random\_number = random.randint(min\_val, max\_val)

# Xuất kết quả ra màn hình

print(f"Số ngẫu nhiên trong đoạn [{min\_val}, {max\_val}] là: {random\_number}")

1. **Khác biệt cơ bản giữa list và tuple?**

**- Cú pháp:** Cú pháp của tập hợp danh sách kiểu List và Tuple là hơi khác nhau. Đối tượng kiểu dữ liệu List được bao quanh bởi dấu ngoặc vuông [] và Tuples được bao quanh bởi dấu ngoặc tròn ().

**Ví dụ:**

1. list\_num = [1,2,3,4]
2. tup\_num = (1,2,3,4)
3. print(list\_num)
4. print(tup\_num)

**Đói tượng:** Đối tượng kiểu List có tính chất có thể thay đổi, tức là đối tượng kiểu List có thể được thay đổi hoặc sửa đổi sau khi tạo tùy theo nhu cầu trong khi đối tượng kiểu tuple có tính chất bất biến, tức là không thể thay đổi được hoặc sửa đổi tuple sau khi tạo.

**Ví dụ:**

1. list\_num = [1,2,3,4]
2. tup\_num = (1,2,3,4)
3. list\_num[2] = 5
4. print(list\_num)
5. tup\_num[2] = 5

**- Các thao tác:** Đối tượng kiểu List có nhiều hàm bên trong hơn so với đối tượng kiểu tuple. Chúng ta có thể sử dụng hàm dir([object]) để lấy ra tất cả các hàm liên quan cho list và tuple.

**Ví dụ:**

list\_num = [1,2,3,4]

tup\_num = (1,2,3,4)

print(dir(list\_num))

print(dir(tup\_num))

**- Kích thước:** Các thao tác trên đối tượng kiểu Tuples có kích thước nhỏ hơn so với thao tác trong các đối tượng kiểu List, điều này làm cho nó nhanh hơn một chút nhưng không phải là quá nhiều.

**Ví dụ:**

a= (1,2,3,4,5,6,7,8,9,0)

b= [1,2,3,4,5,6,7,8,9,0]

print('a=',a.\_\_sizeof\_\_())

print('b=',b.\_\_sizeof\_\_())

**- Các cách sử dụng:**

+ Việc sử dụng một tập hợp kiểu Tuple thay vì kiểu List có thể cung cấp cho người lập trình một gợi ý rằng không nên thay đổi dữ liệu.

+ Tập hợp kiểu Tuples thường được sử dụng giống như một kiểu từ điển không có khóa để lưu trữ dữ liệu.

**Ví dụ:**

[('Swordfish', 'Dominic Sena', 2001), ('Snowden', ' Oliver Stone', 2016), ('Taxi Driver', 'Martin Scorsese', 1976)]

1. **Ứng dụng kiểu dữ liệu tuple trong thực tế?**

- Lưu trữ các giá trị cố định (immutable)

**Ví dụ:**

# Lưu trữ tọa độ cố định

point = (3, 4)

- Sử dụng làm khóa cho dictionary:

**Ví dụ:**

# Lưu trữ tọa độ và tên của các thành phố

locations = {(40.7128, -74.0060): "New York",

(34.0522, -118.2437): "Los Angeles",

(51.5074, -0.1278): "London"}

- Trả về nhiều giá trị từ một hàm:

**Ví dụ:**

def get\_min\_max(numbers):

return min(numbers), max(numbers)

min\_val, max\_val = get\_min\_max([3, 5, 1, 9, 7])

print(f"Min: {min\_val}, Max: {max\_val}")

- Gán nhiều biến cùng lúc:

**Ví dụ:**

# Gán giá trị cho nhiều biến cùng lúc

a, b, c = (1, 2, 3)

- Lưu trữ cấu trúc dữ liệu phức tạp:

**Ví dụ:**

# Lưu trữ thông tin sinh viên

student\_info = ("Alice", "2000-05-15", "12345678")