1. Phân biệt toán tử định dạng chuỗi và hàm định dạng chuỗi trong Python

-Toán tử định dạng chuỗi (f-string) là một cách mới và hiệu quả để nhúng các biểu thức Python vào bên trong chuỗi. Nó được giới thiệu từ phiên bản Python 3.6.

-Hàm định dạng chuỗi (format()) là một phương thức của đối tượng chuỗi, cho phép bạn định dạng các giá trị bằng cách chèn chúng vào các vị trí giữ chỗ trong chuỗi.

2.

import random

def random\_number(start, end):

return random.randint(start, end)

min\_value = int(input("nhap gia tri nho nhat: "))

max\_value = int(input("nhap gia tri lon nhat: "))

random\_num = random\_number(min\_value, max\_value)

print("Số ngẫu nhiên:", random\_num)

3.

\*Tính bất biến:

List: Có thể thay đổi các phần tử sau khi tạo.

Tuple: Không thể thay đổi các phần tử sau khi tạo (bất biến).

\*Sử dụng:

List: Thường được sử dụng để lưu trữ các tập hợp dữ liệu có thể thay đổi, như danh sách các số, danh sách các chuỗi, v.v.

Tuple: Thường được sử dụng để lưu trữ các dữ liệu không thay đổi, như các hằng số, các cặp khóa-giá trị trong từ điển, v.v.

\*Hiệu suất:

Tuple: Thường nhanh hơn list một chút vì tính bất biến của nó.

\*Tính năng:

List: Hỗ trợ nhiều phương thức như append(), remove(), extend(), pop(), v.v.

Tuple: Không hỗ trợ các phương thức thay đổi, chỉ có các phương thức truy xuất như count() và index().

\*Sử dụng:

List: Thường được sử dụng khi bạn cần một tập hợp các phần tử mà có thể thay đổi hoặc sắp xếp lại.

Tuple: Thường được sử dụng khi bạn cần một tập hợp các phần tử mà không muốn bị thay đổi, chẳng hạn như các phần tử của một hàm trả về nhiều giá trị.

\*Kích thước và bộ nhớ:

List: Do tính thay đổi, list có thể tiêu tốn nhiều bộ nhớ hơn và yêu cầu nhiều xử lý hơn để duy trì cấu trúc dữ liệu.

Tuple: Nhờ tính không thay đổi, tuple có thể sử dụng bộ nhớ hiệu quả hơn và thường có kích thước nhỏ hơn so với list.

4.

-Lưu trữ các hằng số: Ví dụ: các ngày trong tuần, các tháng trong năm.

-Làm khóa cho từ điển: Vì tuple là bất biến nên nó có thể làm khóa cho từ điển.

-Trả về nhiều giá trị từ một hàm: Hàm có thể trả về một tuple chứa nhiều giá trị.

-Đại diện cho các điểm tọa độ: Ví dụ: (x, y) trong đồ họa.

-Truyền nhiều đối số cho một hàm: Có thể truyền một tuple làm đối số cho một hàm.

- Tính Toán Tập Hợp và Xử Lý Dữ Liệu:

* Trong một số ứng dụng khoa học và phân tích dữ liệu, tuples có thể dùng để lưu trữ các giá trị trong một tập hợp dữ liệu không thay đổi. Ví dụ, khi làm việc với các dữ liệu thống kê, bạn có thể sử dụng tuples để lưu trữ các giá trị tham số.

- Xử Lý Dữ Liệu Đa Chiều:

* Tuples rất tiện lợi khi làm việc với dữ liệu đa chiều, chẳng hạn như hình ảnh (x, y, z), hoặc dữ liệu thời gian (ngày, giờ, phút).

- Gán Giá Trị Nhanh:

* Tuples cho phép bạn gán nhiều giá trị vào nhiều biến trong một dòng code, giúp mã nguồn trở nên sạch sẽ và dễ đọc hơn.

- Tuple Unpacking:

* Tuple unpacking là một kỹ thuật phổ biến trong Python để gán nhiều giá trị từ một tuple vào các biến khác. Điều này rất hữu ích khi làm việc với các cấu trúc dữ liệu phức tạp.

- Hàm và Biểu Thức Lambda:

* Khi sử dụng hàm lambda hoặc các hàm cao cấp, tuples có thể được sử dụng để làm đối số cho các hàm mà không cần tạo các lớp hoặc cấu trúc dữ liệu khác.