Câu1: 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?

Phân biệt:

toán tử định dạng chuỗi:

Cú pháp cũ và ít linh hoạt hơn.

Dễ bị lỗi khi có nhiều kiểu dữ liệu hoặc định dạng phức tạp.

Ví dụ: "Name: %s, Age: %d" % (name, age)

**hàm định dạng chuỗi** có sẵn trong gói thư viện chuẩn (bao gồm phương thức format() và f-strings):

Phương thức format():

Cung cấp nhiều tùy chọn hơn với chỉ mục và định dạng.

Được khuyến khích sử dụng trước khi f-strings xuất hiện.

Ví dụ: "Name: {}, Age: {}".format(name, age)

f-strings:

Cung cấp cú pháp ngắn gọn và dễ đọc.

Hỗ trợ biểu thức và các phép toán trong dấu {}.

Khuyến khích sử dụng vì hiệu suất cao và dễ đọc hơn.

Ví dụ: f"Name: {name}, Age: {age}"

Câu2:Viết chương trình xuất ra số ngẫu nhiên trong một đoạn bất kỳ bất cho

trước?

import random

def random\_number(start, end):

return random.randint(start, end)

a = int(input("Nhap so nho nhat : "))

b = int(input("Nhap so lon nhat : "))

print("Số ngẫu nhiên:", random\_number(a, b))

Câu3:Khác biệt cơ bản giữa list và tuple?

1. Tính biến đổi (mutability):

List: Có thể thay đổi sau khi được tạo. Bạn có thể thêm, xóa hoặc sửa đổi các phần tử trong một list.

Tuple: Không thể thay đổi sau khi được tạo. Các phần tử trong tuple được cố định và không thể sửa đổi.

2. Cú pháp:

List: Được bao quanh bởi dấu ngoặc vuông [].

Tuple: Được bao quanh bởi dấu ngoặc tròn ().

3. Sử dụng:

List:

Lưu trữ các tập hợp dữ liệu có thể thay đổi kích thước.

Làm việc với các cấu trúc dữ liệu phức tạp như stack, queue.

Sử dụng trong các thuật toán và cấu trúc dữ liệu.

Tuple:

Lưu trữ các dữ liệu không thay đổi, như các hằng số.

Làm khóa cho dictionary.

Trả về nhiều giá trị từ một hàm.

Đảm bảo tính toàn vẹn của dữ liệu.

Tính năng List Tuple

Tính chất Có thể thay đổi Không thể thay đổi

Cú pháp [] ()

Sử dụng Lưu trữ dữ liệu có thể thay đổi Lưu trữ dữ liệu không đổi, khóa cấu trúc dữ liệu dictionary, trả về nhiều giá trị

Câu4:Ứng dụng kiểu dữ liệu tuple trong thực tế?

một số ứng dụng điển hình của tuple:

1. Lưu trữ dữ liệu không đổi:

Thông tin về người: (tên, tuổi, địa chỉ)

Tọa độ: (x, y, z)

Màu sắc: (red, green, blue)

Các hằng số: (PI, e)

2. Làm khóa cho dictionary:

Vì tuple là không thể thay đổi nên nó có thể được sử dụng làm khóa trong dictionary. Điều này đảm bảo rằng mỗi khóa là duy nhất và không thể bị thay đổi.

3. Trả về nhiều giá trị từ một hàm:

Một hàm trong Python chỉ có thể trả về một giá trị. Tuy nhiên, bằng cách sử dụng tuple, bạn có thể trả về nhiều giá trị cùng một lúc.

### 4. ****Đảm bảo tính toàn vẹn của dữ liệu:****

Khi bạn muốn đảm bảo rằng một số giá trị không bị thay đổi trong suốt quá trình thực hiện chương trình, tuple là một lựa chọn tốt.

5. Các ứng dụng khác:

Định nghĩa các cấu trúc dữ liệu: Tuple có thể được sử dụng để định nghĩa các cấu trúc dữ liệu đơn giản như điểm, vector, ma trận.

Truyền tham số cho các hàm: Tuple có thể được sử dụng để truyền nhiều tham số cho một hàm.

Xử lý dữ liệu: Tuple có thể được sử dụng để lưu trữ các kết quả trung gian trong quá trình xử lý dữ liệu