CÂU HỎI TỔNG HỢP

Câu 1: Phân biệt toán tử định dạng chuỗi và hàm định dạng chuỗi trong Python Toán tử định dạng chuỗi (%):

\* Sử dụng ký tự % để chèn giá trị vào chuỗi

. \* Cú pháp: "chuỗi % (giá trị)". Hàm định dạng chuỗi (str.format()): \* Sử dụng phương thức .format() để chèn giá trị vào chuỗi

. \* Cú pháp: "chuỗi {}.format(giá trị)". name = "Alice" age = 30 print("My name is %s and I am %d years old." % (name, age)) print("My name is {} and I am {} years old.".format(name, age)) pi = 3.14159 print("Pi is approximately %.2f." % pi) print("Pi is approximately {:.2f}.".format(pi)) x, y = 10, 20 print("x = %d, y = %d" % (x, y)) print("x = {}, y = {}".format(x, y)) char = 'A' print("Character: %c" % char) print("Character: {}".format(char)) language = "Python" print("I love %s programming." % language) print("I love {} programming.".format(language))

Câu 2:. Chương trình xuất ra số ngẫu nhiên trong một đoạn bất kỳ cho trước Bạn có thể sử dụng thư viện random trong Python để tạo số ngẫu nhiên. Dưới đây là ví dụ: import random start = 1 end = 100 random\_number = random.randint(start, end) print(f"Số ngẫu nhiên trong đoạn từ {start} đến {end}: {random\_number}")

**Câu 3:Khác biệt cơ bản giữa list và tuple**

1. List: \* Có thể thay đổi (mutable). \* Sử dụng dấu ngoặc vuông []. \* Có thể thêm, xóa, hoặc thay đổi phần tử sau khi tạo.

Ví dụ:my list =[1,2,3]

1. Tuple: \* Không thể thay đổi (immutable). \* Sử dụng dấu ngoặc tròn (). \* Không thể thêm, xóa, hoặc thay đổi phần tử sau khi tạo
2. my\_list.append(4) print(my\_list) my\_tuple = (1, 2, 3) print(my\_tuple)

Câu 4:

**Ứng dụng kiểu dữ liệu tuple trong thực tế** \* Lưu trữ dữ liệu không thay đổi: Tuple thường được sử dụng để lưu trữ dữ liệu mà bạn không muốn thay đổi trong suốt quá trình chạy chương trình, ví dụ như tọa độ địa lý (latitude, longitude). \* Khóa trong từ điển: Tuple có thể được sử dụng làm khóa trong từ điển vì chúng không thể thay đổi. \* Trả về nhiều giá trị từ hàm: Tuple thường được sử dụng để trả về nhiều giá trị từ một hàm. location = (10.762622, 106.660172) print(f"Tọa độ địa lý: {location}"

**Tính bất biến**: List: Làm việc với danh sách, bạn có thể thay đổi các phần tử của nó sau khi đã tạo ra. Ví dụ, bạn có thể thêm, xóa hoặc thay đổi các phần tử trong danh sách. Tuple: Làm việc với tuple, bạn không thể thay đổi các phần tử sau khi đã tạo ra. Tuple là bất biến, nghĩa là nó không thể bị thay đổi.

**Cú pháp**: List: Được khai báo bằng dấu ngoặc vuông []. Ví dụ: my\_list = [1, 2, 3]. Tuple: Được khai báo bằng dấu ngoặc đơn (). Ví dụ: my\_tuple = (1, 2, 3).

Hiệu suất: List: Có thể kém hiệu quả hơn tuple vì tính linh hoạt của nó, như việc thay đổi kích thước và giá trị.

**Tuple**: Thường có hiệu suất tốt hơn so với list khi không cần thay đổi nội dung, vì tuple là bất biến và có cấu trúc đơn giản hơn. Sử dụng: List: Thường được sử dụng khi bạn cần một cấu trúc dữ liệu có thể thay đổi. Tuple: Thường được sử dụng khi bạn cần một tập hợp các giá trị không thay đổi, hoặc khi bạn muốn nhóm các giá trị có ý nghĩa liên quan. Tóm lại, danh sách và tuple có các đặc điểm khác nhau về tính bất biến và cách sử dụng, và việc chọn lựa giữa chúng phụ thuộc vào yêu cầu cụ thể của bạn trong lập trình.