Họ tên: Nguyễn Huy Hiệu - Lớp DHTMDT19D

Mssv: 23705461

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?

2. Viết chương trình xuất ra số ngẫu nhiên trong một đoạn bất kỳ bất cho

trước?

3. Khác biệt cơ bản giữa list và tuple?

4. Ứng dụng kiểu dữ liệu tuple trong thực tế?

Bài làm:

1:

**Định nghĩa:**

1. string (formatted string literal) là một cách để định dạng chuỗi trong Python, được giới thiệu từ phiên bản Python 3.6. F-string cho phép chèn các biểu thức Python trực tiếp vào bên trong chuỗi bằng cách đặt chúng trong cặp dấu ngoặc nhọn `{}` và thêm ký tự `f` hoặc `F` trước chuỗi.

`str.format()` là một phương thức định dạng chuỗi trong Python, cho phép chèn các giá trị hoặc biến vào chuỗi thông qua các chỗ giữ chỗ (placeholders) được đánh dấu bằng dấu ngoặc nhọn `{}`. Phương thức này có mặt từ Python 2.7 và Python 3.0.

**Ví dụ:**

**F-string:**

```python

name = "Alice"

age = 30

greeting = f"My name is {name} and I am {age} years old."

print(greeting)

```

**str.format():**

```python

name = "Alice"

age = 30

greeting = "My name is {} and I am {} years old.".format(name, age)

print(greeting)

**So sánh:**

- Cú pháp: F-string ngắn gọn hơn, dễ đọc hơn vì bạn có thể chèn trực tiếp biến vào chuỗi mà không cần sử dụng phương thức `.format()`.

- Hiệu suất: F-string nhanh hơn, đặc biệt là khi cần định dạng chuỗi nhiều lần hoặc trong các vòng lặp lớn.

- Tính năng: Cả hai đều hỗ trợ định dạng số, chuỗi và nhiều tính năng khác, nhưng f-string có ưu thế khi làm việc với các biểu thức phức tạp hoặc khi bạn cần định dạng nhanh chóng trong một câu lệnh.

Khi nào nên sử dụng

F-string : Khi bạn cần hiệu suất tốt hơn và cú pháp dễ đọc, đặc biệt là trong các phiên bản Python từ 3.6 trở lên.

str.format(): Khi bạn cần hỗ trợ các phiên bản Python cũ hơn hoặc cần sử dụng tính năng định dạng nâng cao mà f-string không cung cấp.

2:

import random

start= 1

end = 100

so = random.randint(start , end)

print("so dc chon la:", so)

3. \*\*Khác biệt cơ bản giữa List và Tuple\*\*

- Tính biến đổi:

+List: Có thể thay đổi (mutable).

+Tuple:Không thể thay đổi (immutable).

- Cú pháp:

+List:Dùng dấu ngoặc vuông `[]`.

+Tuple:Dùng dấu ngoặc tròn `()`.

-Hiệu suất:

+List:Chậm hơn và tốn nhiều bộ nhớ hơn.

+Tuple:Nhanh hơn và chiếm ít bộ nhớ hơn.

4.Ứng dụng kiểu dữ liệu Tuple trong thực tế

- Lưu trữ dữ liệu bất biến: Lưu các giá trị không thay đổi như tọa độ hoặc thông tin cá nhân.

- Sử dụng làm khóa trong từ điển: Tuple có thể được dùng làm khóa do tính bất biến.

- Trả về nhiều giá trị từ hàm: Tuple thường được dùng để trả về nhiều giá trị từ một hàm.