**Bài làm**

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

* **Toán tử định dạng chuỗi:**

Giúp chèn hoặc thay đổi giá trị vào một chuỗi theo ý mình, các toán tử định dạng chuỗi phổ biến trong Python:

* **%s**: chuỗi kí tự

***Ví dụ***: ten=”NhuY”

Print(“Ten toi la %s .“%(ten))

* **%d**: số nguyên:

***Ví dụ:***

tuoi=10

print(“Tuoi cua toi la %d.”%(tuoi))

* **%f**: số thực

***Ví dụ:***

a=3.14159

print(“Gia tri cua a la: %.2f “%(a))

* **Hàm định dạng chuỗi:**

Là các hàm hoặc phương thức giúp định dạng hoặc thay đổi cách hiển thị chuỗi văn bản.

***Ví dụ***: name=”NhuY”

print(“My name is {}”.format(name))

***Ví dụ:*** name= “NhuY”

years=2005

print(”My name is {} and I {} born years”.format(name,years))

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

Import random

a=int(input(“Nhập a”))

b=int(input(“Nhập b”))

c=random.randint(a,b)

print("Số ngẫu nhiên",c)

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

* Cú pháp: list được bao quanh bởi dấu () còn tuple được bao quanh bởi []

Ví dụ: list\_num= [1,2,3,4]

print(list\_num)

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

print(tup\_num)

* Tính chất: list có thể thay đổi được trong khi đó tuple có tính chất bất biến
* Đối tượng: list có nhiều hàm bên trong hơn so với tuple
* Độ dài: list có độ dài thay đổi, tuple có độ dài cố định
* Hàm hỗ trợ: list: .append(), .remove(),...

Tuple: count() và .index().

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

* Key trong từ điển:tuple được sử dụng làm key trong từ điển bởi chúng là immutable.

**Ví dụ:** locations = { (40.7128, -74.0060): "New York"}

* Trả về nhiều giá trị từ hàm: Có thể trả về nhiều giá trị từ một hàm bằng tuple, giúp dễ dàng gói các kết quả liên quan lại với nhau.
* Nhóm các đối tượng không thay đổi: khi cần nhóm các đối tượng không thay đổi lại với nhau để sử dụng trong các thuật toán.
* Nhóm giá trị không thay đổi: Khi cần lưu trữ một tập hợp các giá trị mà không muốn thay đổi sau khi tạo.

**Ví dụ:** như tọa độ của một điểm trong không gian (x, y, z) = (5,15,10)

* Gán giá trị: được sử dụng để gán nhiều biến cùng một lúc, điều này giúp mã nguồn ngắn gọn và dễ hiểu hơn.

**Ví dụ:** x,y,z=3,2,4

* Lưu trữ cấu hình: dùng để lưu trữ các giá trị cấu hình không thay đổi của chương trình.
* Sử dụng trong các phép toán với số lượng cố định: khi cần lưu trữ một số lượng cố định các giá trị mà bạn biết trước.
* **Nhóm dữ liệu có liên quan:** dùng khi nhóm các giá trị liên quan mà không cần thay đổi chúng.

**Ví dụ:** khi xử lý dữ liệu từ các file CSV.