NGUYỄN THỊ NHƯ Ý- DHTMDT19D-23716261

**CÂU HỎI TỔNG HỢP**

**Câu 1:**

**-Phân biệt toán tử định dạng chuỗi và hàm định dạng chuỗi có sẵn**

+Toán tử định dạng chuỗi (`%`)

* Là cách truyền thống để định dạng chuỗi trong Python.
* Sử dụng ký tự `%` để chèn các giá trị vào chuỗi.
* Ít linh hoạt hơn và cú pháp phức tạp khi định dạng nhiều giá trị hoặc đối tượng phức tạp.

+Hàm định dạng chuỗi (`str.format()`)

* Là phương pháp hiện đại hơn, được giới thiệu từ Python 2.7 và 3.x.
* Sử dụng dấu ngoặc nhọn `{}` làm chỗ giữ chỗ trong chuỗi và truyền giá trị vào thông qua hàm `format()`.
* Linh hoạt hơn, dễ đọc và hỗ trợ nhiều tùy chọn định dạng.

**-5 ví dụ minh họa tương ứng:**

1.Toán tử % với chuỗi

ten = "Alice"

ketqua = "Hello, %s!" % ten

print(ketqua)

🡺 "Hello, Alice!"

2. Toán tử % với số nguyên

tuoi = 30

ketqua = "I am %d years old." % tuoi

print(ketqua)

🡺 "I am 30 years old."

3.str.format() với nhiều biến

ten = "Bob"

tuoi = 25

ketqua = "Tôi tên là {} và tôi {} tuổi.".format(ten, tuoi)

print(ketqua)

🡺 ‘Tôi tên là Bob và tôi 25 tuổi

4.str.format với chỉ số

trai1 = ‘táo’

trai2 = ‘chuối’

ketqua = ‘tôi thích {0} hơn {1}’.format(trai1, trai2)

print(ketqua)

🡺tôi thích táo hơn chuối

5.F-strings với biểu thức

a = 5

b = 10

ketqua = f”Tổng của {a} và {b} là {a + b}.”

print(ketqua)

🡺Tổng của 5 và 10 là 15

**Câu 2:**

Import random

so = (“5”,”7”,”33”,”44”,”65”,”77”)

ketqua=random.choice(so)

print(ketqua)

**Câu 3:**

Khác biệt cơ bản giữa list và tuple trong Python:

**-Tính thay đổi:**

+List: Có thể thay đổi, nghĩa là bạn có thể thêm, xóa, hoặc sửa đổi các phần tử sau khi list đã được tạo.

+Tuple: Không thể thay đổi, nghĩa là sau khi tuple được tạo, bạn không thể thay đổi các phần tử bên trong.

**-Cú pháp:**

+List: Được tạo bằng dấu ngoặc vuông [].

+Tuple: Được tạo bằng dấu ngoặc tròn ().

**-Hiệu suất:**

+List: Chậm hơn tuple một chút vì có khả năng thay đổi.

+Tuple: Nhanh hơn list vì tính bất biến, đặc biệt trong việc truy cập dữ liệu.

**- Sử dụng:**

+List: Thích hợp khi bạn cần một cấu trúc dữ liệu có thể thay đổi.

+Tuple: Thích hợp khi bạn cần một cấu trúc dữ liệu cố định hoặc sử dụng như một khóa trong `dictionary`.

**Câu 4:**

Ứng dụng của tuple trong

-Khóa cho từ điển: Đảm bảo tính duy nhất và không bị thay đổi của khóa.

-Lưu trữ dữ liệu cố định: Như tọa độ, ngày tháng.

-Trả về nhiều giá trị: Từ một hàm.

-Gom nhóm dữ liệu: Tạo các cấu trúc dữ liệu có thứ tự.

-Bảo vệ dữ liệu: Ngăn chặn việc sửa đổi vô tình.