Câu 1:

-Toán tử định dạng chuỗi: Toán tử là các biểu tượng cụ thể được thực hiện ở một số hoạt động trên một giá trị và cho ra một kết quả khác, được sử dụng để chèn các giá trị vào trong chuỗi.

-Hàm định dạng chuỗi: Hàm định dạng chuỗi là một kiểu dữ liệu để lưu trữ các ký tự hay một chuỗi các ký tự, cung cấp nhiều tính năng linh hoạt hơn so với toán tử.

Ví dụ

1)

name = "Alice"

age = 30

formatted\_string = "Name: %s, Age: %d" % (name, age)

print(formatted\_string)

Kết quả:

Toán tử: Name: Alice, Age: 30

Hàm định dạng chuỗi: Name: Alice, Age: 25

2)

price = 19.99

formatted\_string = "Price: $%.2f" % price

print(formatted\_string)

Kết quả:

Toán tử: Price: $19.99

Hàm định dạng chuỗi: Price: $19.99

3)

data = (3, 7)

formatted\_string = "Point: x = %d, y = %d" % data

print(formatted\_string)

Kết quả:

Toán tử: Point: x = 3, y = 7

Hàm định dạng chuỗi: Point: x = 3, y = 7

4)

percentage = 75.5

formatted\_string = "Completion: %.1f%%" % percentage

print(formatted\_string)

Kết quả

Toán tử: Completion: 75.5%

Hàm định dạng chuỗi: Completion: 75.5%

5)

hex\_value = 255

formatted\_string = "Hexadecimal: %x" % hex\_value

print(formatted\_string)

Kết quả

Toán tử: Hexadecimal: ff

Hàm định dạng chuỗi: Hexadecimal: ff

Câu 2:

import random

print (random.randrange(1,100,10))

print(random.uniform(10,100))

Câu 3:

List: Là một kiểu dữ liệu có thể thay đổi (mutable), có nghĩa là bạn có thể thay đổi, thêm hoặc xóa các phần tử sau khi danh sách đã được tạo.

Có thể thay đổi. Bạn có thể thay đổi giá trị của các phần tử, thêm phần tử mới hoặc xóa các phần tử hiện có.

Cú pháp: my\_list = [1, 2, 3]

Tuple: Là một kiểu dữ liệu không thay đổi (immutable), có nghĩa là một khi tuple được tạo ra, bạn không thể thay đổi, thêm hoặc xóa các phần tử của nó.

Không thể thay đổi. Một khi tuple được tạo ra, bạn không thể thay đổi, thêm hoặc xóa các phần tử.

Cú pháp: my\_tuple = (1, 2, 3)

Câu 4:

Nhóm Dữ Liệu

Tuple thường được sử dụng để nhóm các giá trị liên quan với nhau. Khi bạn cần nhóm các giá trị có liên quan mà không muốn thay đổi chúng, tuple là lựa chọn lý tưởng.

Trả Về Nhiều Giá Trị Từ Hàm

Khi một hàm cần trả về nhiều giá trị, tuple là cách phổ biến để đóng gói các giá trị này và trả về chúng cùng một lúc.

3. Lưu Trữ Dữ Liệu Không Thay Đổi

Nếu bạn cần lưu trữ dữ liệu mà không muốn thay đổi, tuple là lựa chọn tốt. Các giá trị trong tuple không thể bị thay đổi, giúp đảm bảo tính toàn vẹn của dữ liệu.

4. Dùng làm Khóa trong Từ Điển (Dictionary Keys)

Vì tuple là không thay đổi, nó có thể được sử dụng như một khóa trong từ điển (dictionary). Điều này không thể thực hiện được với list vì list là kiểu dữ liệu thay đổi.

5. Sử Dụng Trong Các Cấu Trúc Dữ Liệu Đặc Biệt

Trong một số cấu trúc dữ liệu, như namedtuple hoặc các lớp dữ liệu, tuple thường được sử dụng để lưu trữ các trường dữ liệu. namedtuple là một tiện ích trong module collections giúp tạo các tuple có tên cho các trường.

6. Chia nhỏ Tuplea

Tuple có thể được sử dụng để trả về các giá trị từ một hàm và sau đó có thể được sử dụng để phân tách thành các biến khác nhau.

### 7. **Những Tình Huống Đặc Biệt**

* **Những Tuples với Một Phần Tử**: Tuple với một phần tử cần có dấu phẩy để phân biệt với giá trị đơn lẻ.
* **Tuple trong Phân Tích Dữ Liệu**: Đôi khi trong các công việc phân tích dữ liệu, các cặp (key, value) hoặc các bộ dữ liệu phức tạp khác có thể được lưu trữ dưới dạng tuple.