**Câu 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?**

**\_Toán tử định dạng chuỗi\_**

Toán tử ‘%’ được sử dụng để chèn các giá trị vào trong chuỗi bằng cách sử dụng các mã định dạng.

#### **Ví dụ:**

# **Ví dụ 1**: Định dạng chuỗi bằng cách chèn một số nguyên

name = "Alice"

age = 25

result = "My name is %s and I am %d years old." % (name, age)

print(result) # Kết quả: My name is Alice and I am 25 years old.

# **Ví dụ 2**: Định dạng số thực với số chữ số thập phân cố định

pi = 3.14159

result = "The value of Pi is approximately %.2f." % pi

print(result) # Kết quả: The value of Pi is approximately 3.14.

# **Ví dụ 3**: Định dạng chuỗi với phần trăm

percentage = 95

result = "You scored %d%% in the exam." % percentage

print(result) # Kết quả: You scored 95% in the exam.

# **Ví dụ 4**: Định dạng chuỗi với số thập phân và số nguyên

quantity = 10 price = 99.99

result = "You bought %d items at $%.2f each." % (quantity, price)

print(result) # Kết quả: You bought 10 items at $99.99 each.

# **Ví dụ 5**: Định dạng số nguyên trong hệ thập lục phân

number = 255

result = "The number %d in hexadecimal is %x." % (number, number)

print(result) # Kết quả: The number 255 in hexadecimal is ff.

**\_Hàm định dạng chuỗi (str.format() và f-strings)\_**

Cung cấp cách định dạng chuỗi hiện đại hơn và linh hoạt hơn.

str.format() cho phép sử dụng các chỉ định vị trí và định dạng linh hoạt.

f-strings cho phép nhúng các biểu thức trực tiếp vào chuỗi và dễ đọc hơn**.**

**Ví dụ**

# **Ví dụ 1**: Định dạng chuỗi với tên và tuổi

0name = "Bob" age = 30

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

print(result) # Kết quả: My name is Bob and I am 30 years old.

# **Ví dụ 2**: Định dạng số thực với số chữ số thập phân cố định

pi = 3.14159

result = "The value of Pi is approximately {:.2f}.".format(pi)

print(result) # Kết quả: The value of Pi is approximately 3.14.

# **Ví dụ 3**: Định dạng chuỗi với phần trăm percentage = 85 result = "You scored {}% in the exam.".format(percentage) print(result) # Kết quả: You scored 85% in the exam.

# **Ví dụ 4**: Định dạng chuỗi với các giá trị được đặt tên

quantity = 5

price = 49.99 result = "You bought {qty} items at ${price:.2f} each.".format(qty=quantity, price=price)

print(result) # Kết quả: You bought 5 items at $49.99 each.

# **Ví dụ 5**: Định dạng số nguyên trong hệ thập lục phân

number = 255

result = "The number {} in hexadecimal is {:x}.".format(number, number)

print(result) # Kết quả: The number 255 in hexadecimal is ff.

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

import random

# Nhập vào đoạn bất kỳ

start = int(input("Nhập vào giới hạn dưới của đoạn: "))

end = int(input("Nhập vào giới hạn trên của đoạn: "))

# Sinh ra một số ngẫu nhiên trong đoạn [start, end]

random\_number = random.randint(start, end)

# Xuất số ngẫu nhiên ra màn hình

print(f"Số ngẫu nhiên được sinh ra trong đoạn [{start}, {end}] là: {random\_number}")

**Câu 3. Khác biệt cơ bản giữa list và tuple?**

Lists là các đối tượng có thể thay đổi, cho phép sửa đổi sau khi tạo.

Ngược lại, tuples là các đối tượng bất biến, không thể sửa đổi sau khi đã được tạo.

Dữ liệu thuộc kiểu list có thể thay đổi được, còn dữ liệu thuộc kiểu tuple thì không.

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

Tuple (bộ giá trị) trong Python là một dãy các phần tử bất kỳ, được sắp xếp theo thứ tự và không thể thay đổi. Nghĩa là, một khi bạn đã tạo ra một tuple, bạn không thể thêm, xóa hoặc sửa đổi các phần tử trong đó.

**Ví dụ**

def calculate\_area(vertices):

# vertices là một tuple chứa 3 điểm, mỗi điểm là một tuple (x, y)

(x1, y1), (x2, y2), (x3, y3) = vertices

# Công thức tính diện tích tam giác từ 3 tọa độ trong không gian 2D

area = 0.5 \* abs(x1\*(y2-y3) + x2\*(y3-y1) + x3\*(y1-y2))

return area

# Tạo ra một tuple lưu trữ tọa độ 3 đỉnh của tam giác

triangle\_vertices = ((0, 0), (4, 0), (4, 3))

# Tính toán diện tích tam giác

area = calculate\_area(triangle\_vertices)

print(f"Diện tích của tam giác là: {area}")

# Cố gắng thay đổi giá trị bên trong tuple sẽ gây lỗi

# triangle\_vertices[0] = (1, 1) # Sẽ gây ra lỗi TypeError vì tuple là bất biến