Nguyễn Tiến Thăng 23682711 Nhóm 3

1. Phân biệt toán tử định dạng chuỗi và hàm định dạng chuỗi trong Python

Toán tử định dạng chuỗi (f-strings) là một cách mới và hiệu quả để định dạng chuỗi trong Python 3.6 trở lên. Cú pháp của nó đơn giản và dễ đọc, cho phép bạn nhúng trực tiếp các biểu thức Python vào chuỗi bằng cách đặt chúng trong dấu ngoặc nhọn {}.

Hàm định dạng chuỗi format() là một phương thức của đối tượng chuỗi, được sử dụng từ các phiên bản Python trước đó. Nó cũng cho phép định dạng chuỗi nhưng cú pháp có phần phức tạp hơn.

Ví dụ:

|  |  |
| --- | --- |
| Toán tử f-strings | Hàm format() |
| name = "Alice" <br> age = 30 <br> print(f"Hello, my name is {name} and I am {age} years old.") | name = "Alice" <br> age = 30 <br> print("Hello, my name is {} and I am {} years old.".format(name, age)) |
| price = 19.99 <br> print(f"The price is ${price:.2f}") | price = 19.99 <br> print("The price is ${:.2f}".format(price)) |
| pi = 3.14159 <br> print(f"The value of pi is {pi:.3f}") | pi = 3.14159 <br> print("The value of pi is {:.3f}".format(pi)) |
| list1 = [1, 2, 3] <br> print(f"My list is {list1}") | list1 = [1, 2, 3] <br> print("My list is {}".format(list1)) |
| person = {'name': 'Bob', 'age': 25} <br> print(f"His name is {person['name']} and he is {person['age']} years old.") | person = {'name': 'Bob', 'age': 25} <br> print("His name is {name} and he is {age} years old.".format(name=person['name'], age=person['name'])) |

Ưu điểm của f-strings:

1. Cú pháp gọn gàng: Dễ đọc và viết hơn.
2. Tính năng mạnh mẽ: Hỗ trợ định dạng phức tạp, biểu thức trực tiếp trong dấu ngoặc nhọn.
3. Hiệu suất cao: Thường nhanh hơn format().

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

Python

import random

def random\_number(start, end):

"""Trả về một số ngẫu nhiên trong khoảng từ start đến end."""

return random.randint(start, end)

# Ví dụ:

min\_value = 1

max\_value = 100

random\_num = random\_number(min\_value, max\_value)

print("Số ngẫu nhiên:", random\_num)

Giải thích:

* import random: Nhập thư viện random để sử dụng các hàm liên quan đến số ngẫu nhiên.
* random.randint(start, end): Hàm này trả về một số nguyên ngẫu nhiên nằm trong khoảng từ start đến end (cả hai số đều được bao gồm).

3. Khác biệt cơ bản giữa list và tuple

1. Tính bất biến:
   1. List: Có thể thay đổi các phần tử sau khi tạo.
   2. Tuple: Không thể thay đổi các phần tử sau khi tạo (bất biến).
2. Sử dụng:
   1. List: Thường được sử dụng để lưu trữ các tập hợp dữ liệu có thể thay đổi, như danh sách các số, từ, đối tượng.
   2. Tuple: Thường được sử dụng để lưu trữ các dữ liệu không thay đổi, như các hằng số, các cặp khóa-giá trị, các kết quả trả về của hàm.

4. Ứng dụng kiểu dữ liệu tuple trong thực tế

1. Lưu trữ các hằng số: Ví dụ: các ngày trong tuần, các tháng trong năm.
2. Trả về nhiều giá trị từ một hàm: Hàm có thể trả về một tuple chứa nhiều giá trị.
3. Làm khóa cho dictionary: Tuple có thể làm khóa cho dictionary vì chúng bất biến.
4. Định nghĩa các điểm tọa độ: Một tuple có thể biểu diễn một điểm trong không gian 2D hoặc 3D.
5. Tạo các cấu trúc dữ liệu phức tạp: Tuple có thể được sử dụng như một phần của các cấu trúc dữ liệu lớn hơn, như các danh sách các tuple.
6. Ví dụ:

Python

# Lưu trữ các ngày trong tuần

days = ('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')

# Trả về nhiều giá trị từ một hàm

def find\_min\_max(numbers):

return min(numbers), max(numbers)

# Làm khóa cho dictionary

person = {'name': 'Alice', 'age': 30}

person\_tuple = ('Alice', 30)