

Cao Ngọc Thanh Thu 23719291

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 (f-string) là một cách mới và hiệu quả để nhúng các biểu thức Python vào bên trong chuỗi. Nó được giới thiệu từ phiên bản Python 3.6.
- Hàm định dạng chuỗi (format()) là một phương thức của đối tượng chuỗi, cho phép bạn định dạng các giá trị bằng cách chèn chúng vào các vị trí giữ chỗ trong chuỗi.

Ví dụ

- Định dạng số:

```
number = 12345.6789
```

```
print("Formatted number: {:.2f}".format(number)) # Kết quả: 12345.68
```

- Định dạng chuỗi:

```
text = "Hello, World!"
```

```
print(f"The text is: {text.upper()}") # Chuyển đổi thành chữ hoa
```

- Định dạng ngày tháng:

```
import datetime
```

```
today = datetime.date.today()
```

```
print(f"Today's date is: {today:%B %d, %Y}") # Định dạng ngày tháng
```

- Định dạng số với dấu phẩy:

```
value = 1234567.89
```

```
print("{:, .2f}".format(value)) # Kết quả: 1,234,567.89
```

- Định dạng căn chỉnh:

```
name = "Alice"
```

```
print("{:<10} is a good girl".format(name)) # Căn trái trong 10 ký tự
```

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

```
import random
```

```
giatrinho = int(input("Nhập giá trị nhỏ nhất: "))
```

```
giatrilon = int(input("Nhập giá trị lớn nhất: "))
```

```
def random_number(giatrinho, giatrilon):
```

```
    return random.randint(giatrinho, giatrilon)
```

```
print("Số ngẫu nhiên:", random_number(giatrinho, giatrilon))
```

3. Khác biệt cơ bản giữa list và tuple?

List và **tuple** là hai kiểu dữ liệu được sử dụng để lưu trữ các phần tử trong Python. Tuy nhiên, chúng có những đặc điểm khác biệt quan trọng.

a. Tính bất biến (mutability):

- **List:** Có thể thay đổi sau khi được tạo. Bạn có thể thêm, xóa hoặc sửa đổi các phần tử trong list.
- **Tuple:** Không thể thay đổi sau khi được tạo. Các phần tử của tuple được cố định.

b. Cú pháp (syntax):

- **List:** Được bao quanh bởi dấu ngoặc vuông [].

- **Tuple:** Được bao quanh bởi dấu ngoặc tròn (), Đối với tuple có một phần tử, bạn cần thêm dấu phẩy sau phần tử.

c. Hiệu suất (performance):

- **List:** Vì list có thể thay đổi, nó thường chậm hơn tuple trong các thao tác truy xuất dữ liệu.
- **Tuple:** Thường nhanh hơn và tiêu tốn ít bộ nhớ hơn list, vì tuple là bất biến và không cần hỗ trợ các thao tác thay đổi.

d. Ứng dụng (usage):

- **List:**
 - + Lưu trữ các phần tử có thể thay đổi theo thời gian.
 - + Làm việc với các cấu trúc dữ liệu động.
 - + Sử dụng trong các thuật toán và cấu trúc dữ liệu như stack, queue, danh sách liên kết.
- **Tuple:**
 - + Lưu trữ các dữ liệu không thay đổi.
 - + Làm khóa cho dictionary.
 - + Trả về nhiều giá trị từ một hàm.
 - + Đảm bảo tính toàn vẹn của dữ liệu.

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

a. Lưu trữ Dữ liệu Đa Giá trị

Tuples thường được sử dụng để lưu trữ nhiều giá trị liên quan với nhau mà không cần thay đổi. Ví dụ, lưu trữ thông tin về một người, bao gồm tên, tuổi và địa chỉ:

b. Trả Về Nhiều Giá Trị từ Hàm

Hàm có thể trả về nhiều giá trị bằng cách sử dụng tuple. Điều này rất hữu ích khi bạn cần trả về nhiều giá trị khác nhau từ một hàm.

c. Dùng Làm Khóa trong Từ Điển

Tuples có thể được sử dụng làm khóa trong từ điển (dict) vì chúng không thay đổi. Điều này cho phép bạn lưu trữ các giá trị dựa trên các khóa phức tạp.

d. Lưu Trữ Dữ Liệu Không Thay Đổi

Tuples có thể được sử dụng để lưu trữ dữ liệu không nên thay đổi trong suốt vòng đời của ứng dụng, chẳng hạn như cấu hình hoặc cài đặt.

e. Sử Dụng Trong Phân Tích Dữ Liệu

Trong các ứng dụng phân tích dữ liệu hoặc xử lý dữ liệu, tuples có thể dùng để lưu trữ các bản ghi hoặc dữ liệu đa dạng mà không cần thay đổi, giúp đảm bảo tính toàn vẹn của dữ liệu.

f. Đại diện cho Cặp Giá Trị

Tuples thường được sử dụng để đại diện cho cặp giá trị như tọa độ trong không gian 2D hoặc 3D.

g. Đảm Bảo Tính Toàn Vẹn của Dữ Liệu

Vì tuples là bất biến, chúng giúp đảm bảo rằng dữ liệu không bị thay đổi ngẫu nhiên, điều này có thể rất quan trọng trong các ứng dụng yêu cầu tính toàn vẹn dữ liệu cao.

j. Nhóm Dữ Liệu Liên Quan

Tuples có thể được dùng để nhóm các giá trị liên quan mà không cần sử dụng các lớp hoặc đối tượng khác, đặc biệt là khi bạn không cần nhiều chức năng hơn ngoài việc lưu trữ dữ liệu.

Nhìn chung, tuples là một công cụ mạnh mẽ và linh hoạt trong Python, phù hợp cho nhiều tình huống khác nhau khi bạn cần lưu trữ và xử lý dữ liệu không thay đổi.