TÁC GIẢ: 8 SYNC

CỘNG ĐỒNG

- Kevin Nguyễn
- Nhóm Chia Sẻ Công Nghệ
- Nhóm BlockChain
- Tiktok: 8 Sync
- Youtube: 8 Sync Dev
- Zalo

KHÓA HỌC:

- Fullstack Python
- Fullstack Nextjs
- Fullstack Android-IOS

Tài liệu sẽ được cập nhật định kì và thông báo trong group nên các bạn chú ý nhen .

BÀI 13. HỆ THỐNG UNICODE TRONG PYTHON

Các ứng dụng phần mềm thường cần hiển thị các thông điệp ra màn hình bằng nhiều ngôn ngữ khác nhau như Tiếng Anh, Tiếng Pháp, Tiếng Nhật, Tiếng Do Thái hoặc Tiếng Hindi. Kiểu chuỗi của Python sử dụng Tiêu chuẩn Unicode để biểu diễn

các ký tự. Điều này giúp cho chương trình có thể làm việc với tất cả các ký tự khác nhau này.

Một ký tự là thành phần nhỏ nhất của một văn bản. 'A', 'B', 'C', v.v., tất cả đều là các ký tự khác nhau. Tương tự như vậy là 'È' và 'Í'. Một chuỗi Unicode là một chuỗi các điểm mã, là các số từ 0 đến 0x10FFFF (tương đương với 1.114.111 dạng thập phân). Chuỗi này của các điểm mã cần được biểu diễn trong bộ nhớ dưới dạng một tập hợp các đơn vị mã, và các đơn vị mã sau đó được ánh xạ thành các byte 8-bit.

MÃ HÓA KÝ TỰ

Một chuỗi các điểm mã được biểu diễn trong bộ nhớ dưới dạng một tập hợp các đơn vị mã, ánh xạ thành các byte 8-bit. Quy tắc cho việc dịch một chuỗi Unicode thành một chuỗi byte được gọi là mã hóa ký tự.

Có ba loại mã hóa: UTF-8, UTF-16 và UTF-32. UTF viết tắt của Unicode Transformation Format.

HỖ TRỢ UNICODE CỦA PYTHON

Từ Python 3.0 trở đi, Python tích hợp sẵn hỗ trợ cho Unicode. Kiểu str chứa các ký tự Unicode, do đó bất kỳ chuỗi nào được tạo bằng cú pháp chuỗi đơn, chuỗi kép hoặc chuỗi ba dấu ngoặc kép đều được lưu trữ dưới dạng Unicode. Mã hóa mặc định cho mã nguồn Python là UTF-8.

Do đó, chuỗi có thể chứa biểu diễn chữ của một ký tự Unicode (3/4) hoặc giá trị Unicode của nó (\u00BE).

VÍ DỤ

< 8 Sync Dev />

```
var = "3/4"
print(var)
var = "\u00BE"
print(var)
```

Đoạn mã trên sẽ tạo ra đầu ra như sau:

```
< 8 Sync Dev />
```

```
3/4
¾
```

VÍ DỤ

Trong ví dụ sau, một chuỗi '10' được lưu trữ bằng các giá trị Unicode của 1 và 0, lần lượt là \u0031 và u0030.

```
< 8 Sync Dev />
```

```
var = "\u0031\u0030"
print(var)
```

Nó sẽ tạo ra đầu ra sau:

```
< 8 Sync Dev />
```

```
10
```

Chuỗi hiển thị văn bản dưới dạng dễ đọc cho con người, và byte lưu trữ các ký tự dưới dạng dữ liệu nhị phân. Mã hóa chuyển đổi dữ liệu từ một chuỗi ký tự thành một chuỗi byte. Giải mã chuyển đổi các byte trở lại thành các ký tự và ký hiệu dễ đọc cho con người. Quan trọng là không nên nhầm lẫn giữa hai phương pháp này. `encode` là một phương thức chuỗi, trong khi `decode` là một phương thức của đối tượng byte của Python.

VÍ DỤ

Trong ví dụ sau, chúng ta có một biến chuỗi mà gồm các ký tự ASCII. ASCII là một phần của bộ ký tự Unicode. Phương thức `encode()` được sử dụng để chuyển đổi nó thành một đối tượng byte.

< 8 Sync Dev />

```
string = "Hello"
tobytes = string.encode('utf-8')
print(tobytes)
string = tobytes.decode('utf-8')
print(string)
```

Phương thức `decode()` chuyển đổi đối tượng byte trở lại thành đối tượng str. Phương thức mã hóa được sử dụng là utf-8.

< 8 Sync Dev />

```
b'Hello'
Hello
```

VÍ DỤ

Trong ví dụ sau, biểu tượng Rupee (₹) được lưu trữ trong biến bằng giá trị Unicode của nó. Chúng tôi chuyển đổi chuỗi thành byte và sau đó trở lại str.

< 8 Sync Dev />

```
string = "\u20B9"
print(string)
tobytes = string.encode('utf-8')
print(tobytes)
```

```
string = tobytes.decode('utf-8')
print(string)
```

Khi bạn thực thi mã trên, nó sẽ tạo ra đầu ra sau:


```
₹
b'\xe2\x82\xb9'
₹
```