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 12. HƯỚNG DẪN ÉP KIỂU TRONG PYTHON

GIỚI THIỆU

Ép kiểu trong lập trình là quá trình chuyển đổi một đối tượng từ một kiểu dữ liệu sang kiểu dữ liệu khác. Trong phần này, chúng ta sẽ tìm hiểu về việc ép kiểu trong lập trình Python.

Ép kiểu trong Python là quá trình chuyển đổi một giá trị của một kiểu dữ liệu sang một kiểu dữ liệu khác. Python hỗ trợ hai loại ép kiểu - ép kiểu ngầm định và ép kiểu tường minh.

ÉP KIỂU NGẦM ĐỊNH TRONG PYTHON

Khi bất kỳ trình biên dịch/ngôn ngữ nào tự động chuyển đổi đối tượng của một kiểu sang kiểu khác, đó được gọi là ép kiểu tự động hoặc ngầm định. Python là một ngôn ngữ có kiểu dữ liệu mạnh mẽ. Nó không cho phép chuyển đổi kiểu tự động giữa các kiểu dữ liệu không liên quan. Ví dụ, một chuỗi không thể được chuyển đổi thành bất kỳ kiểu số nào. Tuy nhiên, một số nguyên có thể được ép kiểu thành số thực. Các ngôn ngữ khác như JavaScript là ngôn ngữ có kiểu dữ liệu yếu, trong đó một số nguyên được ép kiểu thành chuỗi để nối chuỗi.

Lưu ý rằng yêu cầu bộ nhớ cho mỗi kiểu dữ liệu là khác nhau. Ví dụ, một đối tượng số nguyên trong Python chiếm 4 byte bộ nhớ, trong khi một đối tượng số thực cần 8 byte vì phần thập phân của nó. Do đó, trình thông dịch Python không tự động chuyển đổi một số thực sang số nguyên, vì điều này sẽ dẫn đến mất dữ liệu. Ngược lại, số nguyên có thể dễ dàng được chuyển đổi thành số thực bằng cách đặt phần thập phân của nó bằng 0.

Ép kiểu ngầm định từ số nguyên sang số thực xảy ra khi thực hiện bất kỳ phép toán số học nào giữa các toán hạng số nguyên và số thực.

Giả sử chúng ta có một biến số nguyên và một biến số thực:


```
a = 10   # Đối tượng số nguyên
b = 10.5 # Đối tượng số thực
```

Để thực hiện phép cộng giữa chúng, 10 - đối tượng số nguyên được nâng cấp thành 10.0. Đây là một số thực, nhưng tương đương với giá trị số của nó trước đó. Bây giờ chúng ta có thể thực hiện phép cộng của hai số thực.

< 8 Sync Dev />

```
c = a + b
print(c)  # Output: 20.5
```

Trong ép kiểu ngầm định, một đối tượng Python với kích thước byte nhỏ hơn được nâng cấp để phù hợp với kích thước byte lớn hơn của đối tượng khác trong phép toán. Ví dụ, một đối tượng Boolean được nâng cấp trước thành số nguyên, sau đó trở thành số thực, trước khi thực hiện phép cộng với một đối tượng số thực. Trong ví dụ sau, chúng ta cố gắng cộng một đối tượng Boolean với một số thực. Lưu ý rằng True tương đương với 1 và False tương đương với 0.

< 8 Sync Dev />

```
a = True
b = 10.5
c = a + b
print(c)  # Output: 11.5
```

ÉP KIỂU TƯỜNG MINH TRONG PYTHON

Mặc dù ép kiểu tự động hoặc ngầm định chỉ giới hạn ở việc chuyển đổi từ số nguyên sang số thực, bạn có thể sử dụng các hàm tích hợp sẵn của Python như int(), float() và str() để thực hiện các chuyển đổi tường minh như chuỗi thành số nguyên.

HÀM INT() TRONG PYTHON

Hàm int() tích hợp sẵn của Python chuyển đổi một hằng số số nguyên thành một đối tượng số nguyên, một số thực thành số nguyên, và một chuỗi thành số nguyên nếu chuỗi đó có biểu diễn số nguyên hợp lệ.

Nếu đối số cho hàm int() là một đối tượng số thực hoặc biểu thức số thực, nó trả về một đối tượng số nguyên. Ví dụ:

```
< 8 Sync Dev />

a = int(10.5)     # Chuyển đổi một đối tượng số thực thành số nguyên
print(a)          # Output: 10
```

Nếu đối số cho hàm int() là một đối tượng Boolean, giá trị trả về là số nguyên 1.

```
< 8 Sync Dev />

a = int(True)     # Chuyển đổi một đối tượng Boolean thành số nguyên
print(a)          # Output: 1
```

Chuyển đổi từ chuỗi thành số nguyên chỉ xảy ra nếu chuỗi đó chứa biểu diễn số nguyên hợp lệ.

```
< 8 Sync Dev />

a = int("100")    # Chuyển đổi một chuỗi thành số nguyên
print(a)          # Output: 100
```

Nếu chuỗi chứa một biểu diễn không hợp lệ của số nguyên, Python sẽ raise một ValueError.

```
< 8 Sync Dev />

a = int("10.5")   # Lỗi: chuỗi không hợp lệ cho int()
```

HÀM FLOAT() TRONG PYTHON

Hàm float() của Python chuyển đổi một hằng số số nguyên thành một đối tượng số thực, một số thực thành một số thực, và một chuỗi thành một số thực nếu chuỗi đó có biểu diễn số thực hợp lệ.

Nếu đối số cho hàm float() là một đối tượng số nguyên, nó trả về một đối tượng số thực.

< 8 Sync Dev />

```
a = float(100)    # Chuyển đổi một đối tượng số nguyên thành số thực
print(a)          # Output: 100.0
```

Nếu đối số cho hàm float() là một chuỗi, và chuỗi đó chứa một biểu diễn số thực hợp lệ, nó sẽ trả về một đối tượng số thực.

< 8 Sync Dev />

```
a = float("10.5") # Chuyển đổi một chuỗi thành số thực
print(a)          # Output: 10.5
```

Nếu chuỗi chứa một biểu diễn không hợp lệ của số thực, Python sẽ raise một ValueError.

< 8 Sync Dev />

```
a = float("Hello") # Lỗi: chuỗi không hợp lệ cho float()
```

HÀM STR() TRONG PYTHON

Hàm str() của Python chuyển đổi một đối tượng thành một biểu diễn chuỗi của đối tượng đó.

< 8 Sync Dev />

```
a = str(10)      # Chuyển đổi một số nguyên thành chuỗi
print(a)         # Output: '10'
```

< 8 Sync Dev />

```
a = str(10.5)    # Chuyển đổi một số thực thành chuỗi
print(a)         # Output: '10.5'
```

< 8 Sync Dev />

```
a = str(True)    # Chuyển đổi một Boolean thành chuỗi
print(a)         # Output: 'True'
```

Hàm str() cũng có thể chuyển đổi một danh sách hoặc một bộ thành một chuỗi.

< 8 Sync Dev />

```
a = str([1, 2, 3])     # Chuyển đổi một danh sách thành chuỗi
print(a)               # Output: '[1, 2, 3]'
```

< 8 Sync Dev />

```
a = str((1, 2, 3))     # Chuyển đổi một bộ thành chuỗi
print(a)               # Output: '(1, 2, 3)'
```

KẾT LUẬN

Trong Python, ép kiểu là một phần quan trọng của việc xử lý dữ liệu. Bằng cách sử dụng các hàm tích hợp sẵn như int(), float() và str(), bạn có thể chuyển đổi giữa các kiểu dữ liệu một cách dễ dàng và linh hoạt. Điều này cho phép bạn làm việc với dữ liệu một cách hiệu quả trong các ứng dụng Python của mình.