

Python - Operadores de Identidade

Operadores de identidade Python

Os operadores de identidade comparam os objetos para determinar se eles compartilham a mesma memória e se referem ao mesmo tipo de objeto (**tipo de dados**).

Python forneceu dois operadores de identidade; nós os listamos da seguinte forma:

- ' **é** ' Operador
- ' **não é** ' Operador

Operador Python 'é'

O operador ' **is** ' é avaliado como True se ambos os objetos operando compartilham o mesmo local de memória. A localização da memória do objeto pode ser obtida pela função "id()". Se o "id()" de ambas as variáveis for igual, o operador "is" retornará True .

Exemplo de operador Python Identity 'is'

```
a = [1, 2, 3, 4, 5]
b = [1, 2, 3, 4, 5]
c = a

# Comparing and printing return values
print(a is c)
print(a is b)

# Printing IDs of a, b, and c
print("id(a) : ", id(a))
print("id(b) : ", id(b))
print("id(c) : ", id(c))
```

Ele produzirá a seguinte **saída** -



```
True
False
id(a) : 140114091859456
id(b) : 140114091906944
id(c) : 140114091859456
```

Operador Python 'não é'

O operador ' **is not** ' é avaliado como True se ambos os objetos operandos não compartilham o mesmo local de memória ou se ambos os operandos não são os mesmos objetos.

Exemplo de operador Python Identity 'is not'

```
a = [1, 2, 3, 4, 5]
b = [1, 2, 3, 4, 5]
c = a

# Comparing and printing return values
print(a is not c)
print(a is not b)

# Printing IDs of a, b, and c
print("id(a) : ", id(a))
print("id(b) : ", id(b))
print("id(c) : ", id(c))
```

Ele produzirá a seguinte **saída** -

```
False
True
id(a) : 140559927442176
id(b) : 140559925598080
id(c) : 140559927442176
```

Exemplos de operadores de identidade Python com explicações

Exemplo 1

```
a="TutorialsPoint"
b=a
print ("id(a), id(b):", id(a), id(b))
print ("a is b:", a is b)
print ("b is not a:", b is not a)
```

Ele produzirá a seguinte **saída** -

```
id(a), id(b): 2739311598832 2739311598832
a is b: True
b is not a: False
```

Os objetos **lista** e **tupla** se comportam de maneira diferente, o que pode parecer estranho à primeira vista. No exemplo a seguir, duas listas "a" e "b" contêm os mesmos itens. Mas o id() deles é diferente.

Exemplo 2

```
a=[1,2,3]
b=[1,2,3]
print ("id(a), id(b):", id(a), id(b))
print ("a is b:", a is b)
print ("b is not a:", b is not a)
```

Ele produzirá a seguinte **saída** -

```
id(a), id(b): 1552612704640 1552567805568
a is b: False
b is not a: True
```

A lista ou tupla contém apenas os locais de memória de itens individuais e não os itens em si. Portanto, "a" contém os endereços de 10,20 e 30 objetos inteiros em um determinado local que pode ser diferente de "b".

Exemplo 3

```
print (id(a[0]), id(a[1]), id(a[2]))
print (id(b[0]), id(b[1]), id(b[2]))
```

Ele produzirá a seguinte **saída** -

```
140734682034984 140734682035016 140734682035048  
140734682034984 140734682035016 140734682035048
```

Devido a duas localizações diferentes de "a" e "b", o operador "**is**" retorna False mesmo que as duas listas contenham os mesmos números.