Identity Operator

To compare the memory locations of two objects

Table of contents

01

Building Understanding

Building Understanding

Identity operators are used to compare the **memory locations** of two objects. These operators help determine whether two variables **point to the same object in memory**, which is different from comparing their values.

There are two identity operators in Python:

- **is**: Evaluates to **True** if both variables are **same object**.

- **is not**: Evaluates to **True** if both variables are **different objects**.

What is “Identity” Exactly?

“**Identity**" refers to the **unique identifier** assigned to an object in **memory**. Every object in Python has a unique identity, which can be thought of as the **memory address** where that object is stored. It can be accessed using “**id()**”. This identity is constant during the object’s lifetime.

Two variables can have the **same value** but **different identities**. This means that while the contents of the variables might be identical, they could still be different objects stored at different memory locations.

02

is Operator

is Operator

is checks whether two variables point to the same object in memory.

Syntax:
x is y

Example:

```python
a = [1, 2, 3]
b = a
c = [1, 2, 3]

print(a is b)  # Output: True
print(a is c)  # Output: False
```

Explanation:

- In this example, **a** and **b** refer to the same list object in memory, so **a is b** returns **True**.
- Although **a** and **c** contain the same elements (and thus have the same value), they are two different objects in memory, so **a is c** returns **False**.

03

is not Operator

is not Operator

is not checks whether two variables point to different objects in memory.

Syntax:
x is not y

Example:

```python
a = [1, 2, 3]
b = [1, 2, 3]

print(a is not b)  # Output: True
```

Explanation:

Even though **a** and **b** have the same value, they are different objects in memory, so **a is not b** returns **True**..

Python Optimization

Integer Caching:

```python
a = 10
b = 10

print(a is b)  # Output: True
```

String Interning

```python
a = "hello"
b = "hello"

print(a is b)  # Output: True
```

Note:

These Optimization Techniques is done on small numbers and strings (or most commonly used strings like keyword or simple text).

If the number or string is large or complex, then memory locations might be different.

04

Identity vs Equality

Identity Vs Equality

Identity operators (**is, is not**) and equality operators **(==, !=)** are DIFFERENT:

Equality (==, !=): These operators check whether the **values** of two objects are the same.

Identity (is, is not): These operators check whether two **variables** point to the same object in memory.

Identity Vs Equality

Example:

Python

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

print(a == b)  # Output: True
print(a == c)  # Output: True

print(a is b)  # Output: True
print(a is c)  # Output: False
```

Explanation:

a == b and **a == c** both return **True** because the values are the same.
a is b returns **True** because **a** and **b** point to the same object.
a is c returns **False** because **a** and **c** are different objects, despite having the same value.

05

Common Mistakes

Mistake 1: Confusing Identity with Equality

Beginners often mistakenly use **is** instead of **==** when they actually want to check for equality.

Example:

Python

```
a = "hello"
b = "hello"

print(a is b)  # Output: True or
False (depends on string interning)
print(a == b)  # Output: True
```

Explanation:

While **a == b** will always be **True** because the values are the same, **a is b** might not always be **True** because the two strings might not refer to the same object in memory.

Mistake 2: Identity Comparisons with Mutable Objects

With mutable objects like lists or dictionaries, identity comparisons can yield unexpected results.

Example:

Python

```
a = [1, 2, 3]
b = [1, 2, 3]

print(a is b)  # Output: False
print(a == b)  # Output: True
```

Explanation:

Since lists are mutable, Python does not optimize their identity in the same way it does for small integers or interned strings. As a result, **a is b** returns **False**.

Knowledge Reinforcement

1

Question

Which of the following operators checks whether two variables refer to the same object in memory?

Answer

a) ==

b) !=

c) is

d) in

1

Question

Which of the following operators checks whether two variables refer to the same object in memory?

Answer

a) ==

b) !=

c) is

d) in

2

Question

What will be the output of the following code?

Python

```
a = [1, 2, 3]
b = a
print(a is b)
```

Answer

a) True

b) False

c) Error

d) None

2

Question

What will be the output of the following code?

Python

```
a = [1, 2, 3]
b = a
print(a is b)
```

Answer

a) True

b) False

c) Error

d) None

3

Question

Which of the following statement is **True** regarding identity operators?

Answer

a) **is** checks for value equality.

b) **is not** checks if two variables have the same value.

c) **is** checks whether two variables refer to the same object in memory.

d) **is not** compares the values stored in two objects.

3

Question

Which of the following statement is **True** regarding identity operators?

Answer

a) **is** checks for value equality.

b) **is not** checks if two variables have the same value.

c) is checks whether two variables refer to the same object in memory.

d) **is not** compares the values stored in two objects.

4

Question

What will be the output of the following code?

Python

```
a = 10
b = 10
print(a is b)
```

Answer

a) True

b) False

c) Error

d) None

4

Question

What will be the output of the following code?

Python

```
a = 10
b = 10
print(a is b)
```

Answer

a) True

b) False

c) Error

d) None

5

Question

What will be the output of the following code?

Python

```
x = [1, 2, 3]
y = [1, 2, 3]
print(x is y)
```

Answer

a) True

b) False

c) Error

d) None

5

Question

What will be the output of the following code?

Python

```
x = [1, 2, 3]
y = [1, 2, 3]
print(x is y)
```

Answer

a) True

b) False

c) Error

d) None

6

Question

Which of the following statements is **True**?

Answer

a) **is** compares values, while **==** compares object identity.

b) **is** and **==** are completely interchangeable.

c) **is** compares object identity, while **==** compares values.

d) **is** not and **!=** perform the same operation.

6

Question

Which of the following statements is **True**?

Answer

a) **is** compares values, while **==** compares object identity.

b) **is** and **==** are completely interchangeable.

c) is compares object identity, while == compares values.

d) **is** not and **!=** perform the same operation.

7

Question

What will be the output of the following code?

Python

```
a = "hello"
b = "hello"
print(a is b)
```

Answer

a) True

b) False

c) Error

d) None

7

Question

What will be the output of the following code?

Python

```
a = "hello"
b = "hello"
print(a is b)
```

Answer

a) True

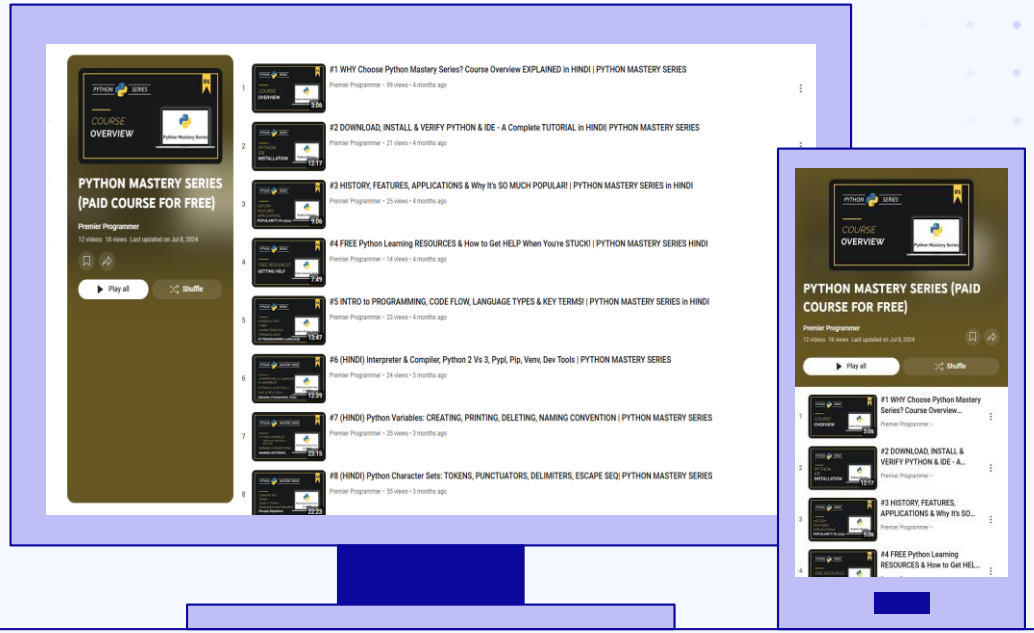
b) False

c) Error

d) None

WATCH

Level up your coding with each episode in this focused Python series.



Next Video!

Walrus Operator-
In Depth