

Membership Operator



Table of contents

01

In Operator

02

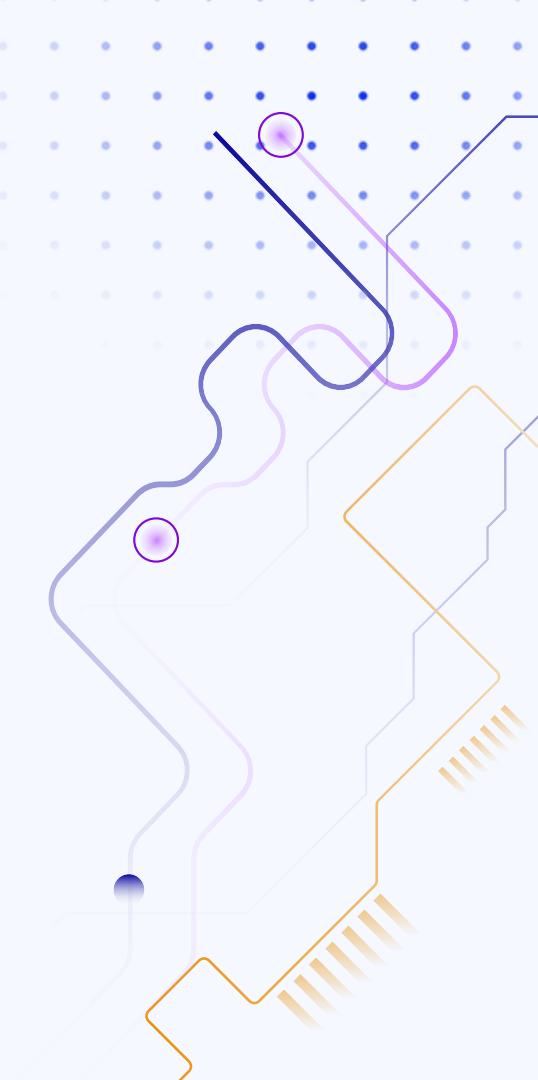
Not in Operator

03

Practical Example

01

In Operator



In Operator

The **in** operator checks if a particular element exists in a sequence. If the element is found, it returns **True**; otherwise, it returns **False**.

Syntax:

element in sequence

Example 1: Checking Membership in a List

Python

```
my_list = [1, 2, 3, 4, 5]
print(3 in my_list) # Output: True
print(6 in my_list) # Output: False
```

In this example, the first check returns **True** because **3** is in **my_list**, while the second check returns **False** because **6** is not in **my_list**.

Example 2: Checking Membership in a String

Python

```
my_string = "Hello, World!"  
print("Hello" in my_string) # Output: True  
print("hello" in my_string) # Output: False
```

Here, the first check is **True** because "Hello" is a substring of **my_string**. The second check is **False** because Python is case-sensitive, and "hello" (with a lowercase 'h') does not match any part of **my_string**.

02

Not In Operator

Not In Operator

The **not in** operator is the opposite of the **in** operator. It checks if an element is not present in a sequence. If the element is not found, it returns **True**; otherwise, it returns **False**.

Syntax:

Element not in sequence

Example 1: Checking Non-Membership in a List

Python

```
my_list = [1, 2, 3, 4, 5]
print(3 not in my_list) # Output: False
print(6 not in my_list) # Output: True
```

In this case, the first check returns **False** because **3** is in **my_list**, while the second check returns **True** because **6** is not in **my_list**.

Example 2: Checking Non-Membership in a String

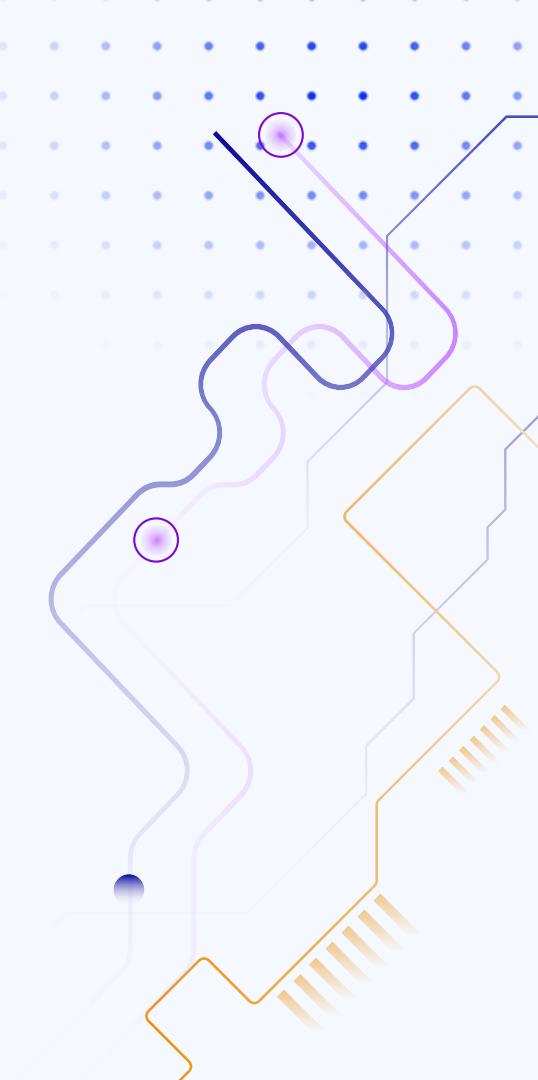
Python

```
my_string = "Hello, World!"  
print("hello" not in my_string) # Output: True  
print("World" not in my_string) # Output: False
```

The first check is **True** because "hello" is not found in **my_string**, while the second check is **False** because "World" is present.

03

Practical Example



User Input Validation

They can also be used to validate user input by checking if the input is part of an allowed set of values.

Python

```
allowed_responses = ['yes', 'no']
response = input("Do you want to continue? (yes/no): ")
if response in allowed_responses:
    print("Valid response.")
else:
    print("Invalid response.")
```

Checking Keys in Dictionaries

In dictionaries, the `in` operator checks for the presence of keys.

Python

```
my_dict = {'name': 'Alice', 'age': 25}  
print('name' in my_dict) # Output: True  
print('address' in my_dict) # Output: False
```

Important Points to Remember

01

Case Sensitivity

Membership checks in strings are case-sensitive.
"Python" and "python" would be considered
different when using **in** or **not in**.

02

Works with Different
Sequences

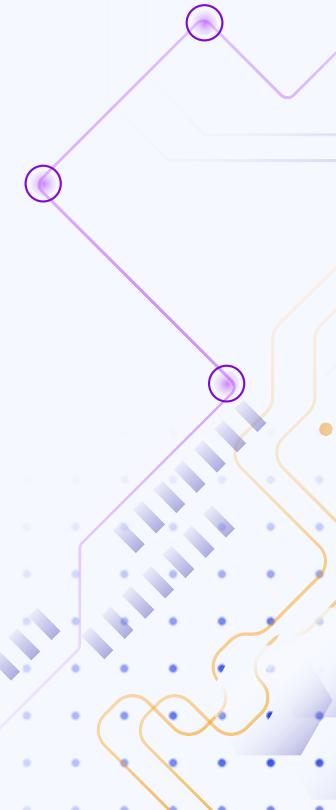
The **in** and **not in** operators can be used with
strings, lists, tuples, dictionaries, and sets.

03

Efficiency

Membership checks can be very efficient with
sets and dictionaries.

Knowledge Reinforcement



1

Question

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

Answer

- a) 4 in [1, 2, 3]
- b) "hello" in "Hello, World!"
- c) "apple" in ["banana", "apple", "cherry"]
- d) 10 not in (10, 20, 30)

1

Question

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

Answer

- a) 4 in [1, 2, 3]
- b) "hello" in "Hello, World!"
- c) "apple" in ["banana", "apple", "cherry"]**
- d) 10 not in (10, 20, 30)

2

Question

What will be the output of the following code?

Python

```
my_dict = {'a': 1, 'b': 2, 'c': 3}  
print('b' in my_dict)
```

Answer

- a) True
- b) False
- c) Error
- d) None

2

Question

What will be the output of the following code?

Python

```
my_dict = {'a': 1, 'b': 2, 'c': 3}  
print('b' in my_dict)
```

Answer

- a) True
- b) False
- c) Error
- d) None

3

Question

Given the list **fruits = ["apple", "banana", "cherry"]**, which of the following statements is **False**?

Answer

- a) "apple" in fruits
- b) "grape" not in fruits
- c) "banana" not in fruits
- d) "cherry" in fruits

3

Question

Given the list **fruits = ["apple", "banana", "cherry"]**, which of the following statements is **False**?

Answer

- a) "apple" in fruits
- b) "grape" not in fruits
- c) "banana" not in fruits**
- d) "cherry" in fruits

4

Question

What will be the output of the following code?

Python

```
message = "Python is fun!"  
print("Fun" in message)
```

Answer

- a) True
- b) False
- c) None
- d) Error

4

Question

What will be the output of the following code?

Python

```
message = "Python is fun!"  
print("Fun" in message)
```

Answer

- a) True
- b) False**
- c) None
- d) Error

5

Question

Which of the following expressions evaluates to **True**?

Answer

- a) "dog" not in "doghouse"
- b) 10 in [5, 15, 25]
- c) "key" in {'key': 'value'}
- d) "e" in "Python"

5

Question

Which of the following expressions evaluates to **True**?

Answer

- a) "dog" not in "doghouse"
- b) 10 in [5, 15, 25]
- c) "key" in {'key': 'value'}**
- d) "e" in "Python"

6

Question

What will be the output of the following code?

Python

```
numbers = [1, 2, 3, 4, 5]
if 6 not in numbers:
    print("6 is not in the list")
else:
    print("6 is in the list")
```

Answer

- a) 6 is not in the list
- b) 6 is in the list
- c) Error
- d) None

6

Question

What will be the output of the following code?

Python

```
numbers = [1, 2, 3, 4, 5]
if 6 not in numbers:
    print("6 is not in the list")
else:
    print("6 is in the list")
```

Answer

- a) 6 is not in the list
- b) 6 is in the list
- c) Error
- d) None

7

Question

Which of the following will return **False**?

Answer

- a) "a" in "apple"
- b) 10 not in [10, 20, 30]
- c) "dog" in ["cat", "dog", "mouse"]
- d) "y" not in "Python"

7

Question

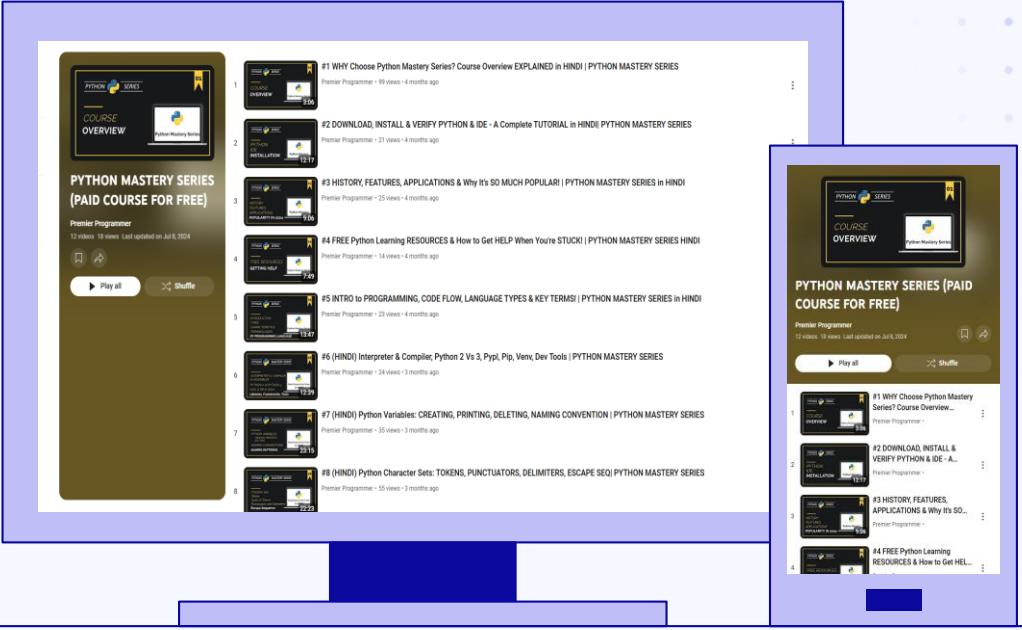
Which of the following will return **False**?

Answer

- a) "a" in "apple"
- b) 10 not in [10, 20, 30]**
- c) "dog" in ["cat", "dog", "mouse"]
- d) "y" not in "Python"

WATCH

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



Next Video!

Identity Operator-
In Depth

