Check if element exists in list in Python

Difficulty Level : Easy • Last Updated : 22 Feb, 2023

Read Discuss Courses Practice Video

The list is an important container in python as it stores elements of all the data types as a collection. Knowledge of certain list operations is necessary for day-day programming. This article discusses the Fastest way to check if a value exists in a list or not using Python.

Example:

```
list = test_list = [1, 6, 3, 5, 3, 4]
Input: 3  # Check if 3 exist or not.
Output: True
```

```
Input: 7  # Check if 7 exist or not.
Output: False
```

Method 1: Naive Method

In the Naive method, one easily uses a loop that iterates through all the elements to check the existence of the target element. This is the simplest way to check the existence of the element in the list. Python is the most conventional way to check if an element exists in a list or not. This particular way returns True if an element exists in the list and False if the element does not exist in the list. The list need not be sorted to practice this approach of checking.

Example 1: Check if an element exists in the list using the if-else statement

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !

Python3

```
# python code to Check if element exists in list or not

lst=[ 1, 6, 3, 5, 3, 4 ]
#checking if element 7 is present
# in the given list or not
i=7
# if element present then return
# exist otherwise not exist
if i in lst:
    print("exist")
else:
    print("not exist")


    # this code is contributed by gangarajula laxmi
```

Output

```
not exist
```

Time Complexity: O(1)

Auxiliary Space: O(n), where n is total number of elements.

Example 2: Check if an element exists in the list using a loop

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Python3

```python
# Initializing list
test_list = [1, 6, 3, 5, 3, 4]

# Checking if 4 exists in list
for i in test_list:
    if(i == 4):
        print("Element Exists")
```

Output:

```
Element Exists
```

Time Complexity: O(n)

Auxiliary Space: O(1)

Example 3: Check if an element exists in the list using "in"

Python3

```python
# Initializing list
test_list = [1, 6, 3, 5, 3, 4]

# Checking if 4 exists in list
# using in
if (4 in test_list):
    print("Element Exists")
```

Output:

```
Element Exists
```

Example 4: Check if an element exists in the list using any() function

Python3

```python
# Initializing list
test_list = [1, 6, 3, 5, 3, 4]
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Output:

```
Does string contain any list element : True
```

Method 2: Check if an element exists in the list using count()

We can use the in-built python List method, count(), to check if the passed element exists in the List. If the passed element exists in the List, the count() method will show the number of times it occurs in the entire list. If it is a non-zero positive number, it means an element exists in the List. Demonstrating to check the existence of elements in the list using count().

Python3

```
# Initializing list
test_list = [10, 15, 20, 7, 46, 2808]

print("Checking if 15 exists in list")

# number of times element exists in list
exist_count = test_list.count(15)

# checking if it is more than 0
if exist_count > 0:
    print("Yes, 15 exists in list")
else:
    print("No, 15 does not exists in list")
```

Output:

```
Checking if 15 exists in list
Yes, 15 exists in list
```

Method 3: Check if an element exists in the list using sort + bisect_left + set

Converting the list into the set and then using *it* can possibly be more efficient than only using it. But having efficiency for a plus also has certain negatives. One among them is that the order of the list is not preserved, and if you opt to take a new list for it, you would require to use extra space. Another drawback is that set disallows duplicity and hence duplicate elements would be removed from the original list. In the conventional binary search way of

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

element ordering. bisect_left() returns the first occurrence of the element to be found and has worked similarly to lower_bound() in C++ STL.

> ***Note:*** *The bisect function will only state the position of where to insert the element but not the details about if the element is present or not.*

Demonstrating to check existence of element in list using set() + in and sort() + bisect_left()

Python3

```
from bisect import bisect_left ,bisect

# Initializing list
test_list_set = [ 1, 6, 3, 5, 3, 4 ]
test_list_bisect = [ 1, 6, 3, 5, 3, 4 ]

print("Checking if 4 exists in list ( using set() + in) : ")

# Checking if 4 exists in list
# using set() + in
test_list_set = set(test_list_set)
if 4 in test_list_set :
    print ("Element Exists")

print("Checking if 4 exists in list ( using sort() + bisect_left() ) : ")

# Checking if 4 exists in list
# using sort() + bisect_left()
test_list_bisect.sort()
if bisect_left(test_list_bisect, 4)!=bisect(test_list_bisect, 4):
    print ("Element Exists")
else:
    print("Element doesnt exist")
```

Output:

```
Checking if 4 exists in list ( using set() + in) :
Element Exists
Checking if 4 exists in list ( using sort() + bisect_left() ) :
Element Exists
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Method 4: Using find() method

Python3

```
# Initializing list
test_list = [10, 15, 20, 7, 46, 2808]

print("Checking if 15 exists in list")
x=list(map(str,test_list))
y="-".join(x)

if y.find("15") !=-1:
    print("Yes, 15 exists in list")
else:
    print("No, 15 does not exists in list")
```

Output

```
Checking if 15 exists in list
Yes, 15 exists in list
```

Method 5: Using Counter() function

Below is the implementation:

Python3

```
from collections import Counter

test_list = [10, 15, 20, 7, 46, 2808]

# Calculating frequencies
frequency = Counter(test_list)

# If the element has frequency greater than 0
# then it exists else it doesn't exist
if(frequency[15] > 0):
    print("Yes, 15 exists in list")
else:
    print("No, 15 does not exists in list")

# This code is contributed by vikkycirus
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

```
Yes, 15 exists in list
```

Method 6: Using try-except block

One additional approach to check if an element exists in a list is to use the index() method. This method returns the index of the first occurrence of the element in the list, or throws a ValueError if the element is not present in the list. To use this method, you can wrap the call to index() in a try-except block to catch the ValueError and return False if it occurs:

Python3

```python
def element_exists(lst, element):
  # Try to get the index of the element in the list
  try:
    lst.index(element)
  # If the element is found, return True
    return True
  # If a ValueError is raised, the element is not in the list
  except ValueError:
  # Return False in this case
    return False

#Test the function
test_list = [1, 6, 3, 5, 3, 4]

print(element_exists(test_list, 3)) # prints True
print(element_exists(test_list, 7)) # prints False
#This code is contributed by Edula Vinay Kumar Reddy
```

Output

```
True
False
```

Time complexity: O(n), where n is the length of the list. The index() method iterates through the list to find the element, so the time complexity is linear.

Auxiliary Space: O(1). This approach does not require any additional space.

Approach 7: Using Set

Time complexity: O(1) average case as checking for an element in a set takes constant time

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

on average.

Space complexity: O(n) as it creates a new set from the list to store its elements.

Python3

```python
def check_element_exists_set(lst, target):
  return target in set(lst)

#Example
test_list = [1, 6, 3, 5, 3, 4]
target = 3
print("Exists using set: ", check_element_exists_set(test_list, target))
```

Output

```
Exists using set:  True
```

87

Related Articles

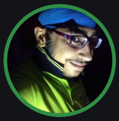
1. Python | Check if element exists in list of lists
2. Python | Check if a list exists in given list of lists
3. Python - Test if any set element exists in List
4. Check if a value exists in a DataFrame using in & not in operator in Python-Pandas
5. Check if Table Exists in SQLite using Python
6. Check if a string exists in a PDF file in Python
7. Check whether given Key already exists in a Python Dictionary
8. How to check if a Python variable exists?
9. Python - Check if a file or directory exists

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

10. Python | Check if tuple exists as dictionary key

Previous Next

Article Contributed By :



manjeet_04
@manjeet_04

Vote for difficulty

Current difficulty : Easy

Easy Normal Medium Hard Expert

Improved By : nidhi_biet, vikkycirus, kogantibhavya, simmytarika5, anantapodder, shreyaperla, surajkr_gupta, laxmigangarajula03, noviced3vq6, mcpe4xwo2, guptapratik, davuluriaakashnag, mallelagowtalm

Article Tags : Python list-programs, python-list, Python

Practice Tags : python, python-list

Improve Article Report Issue

GeeksforGeeks

A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company Languages

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

In Media

Contact Us

Privacy Policy

Copyright Policy

Third-Party Copyright Notices

Advertise with us

C++

GoLang

SQL

R Language

Android Tutorial

Data Structures

Array

String

Linked List

Stack

Queue

Tree

Graph

Algorithms

Sorting

Searching

Greedy

Dynamic Programming

Pattern Searching

Recursion

Backtracking

Web Development

HTML

CSS

JavaScript

Bootstrap

ReactJS

AngularJS

NodeJS

Write & Earn

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

Video Internship

Computer Science

GATE CS Notes

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning Tutorial

Maths For Machine Learning

Pandas Tutorial

NumPy Tutorial

NLP Tutorial

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Company Preparation

Preparation for SDE

Company Interview Corner

Experienced Interview

Internship Interview

Competitive Programming

Aptitude

Python Tutorial

Python Programming Examples

Django Tutorial

Python Projects

Python Tkinter

OpenCV Python Tutorial

GfG School

CBSE Notes for Class 8

CBSE Notes for Class 9

CBSE Notes for Class 10

CBSE Notes for Class 11

CBSE Notes for Class 12

English Grammar

UPSC/SSC/BANKING

SSC CGL Syllabus

SBI PO Syllabus

IBPS PO Syllabus

UPSC Ethics Notes

UPSC Economics Notes

UPSC History Notes

@geeksforgeeks , Some rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy