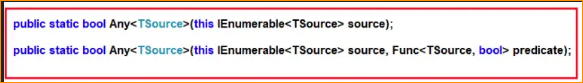
1. What is LINQ Any Method in C#?

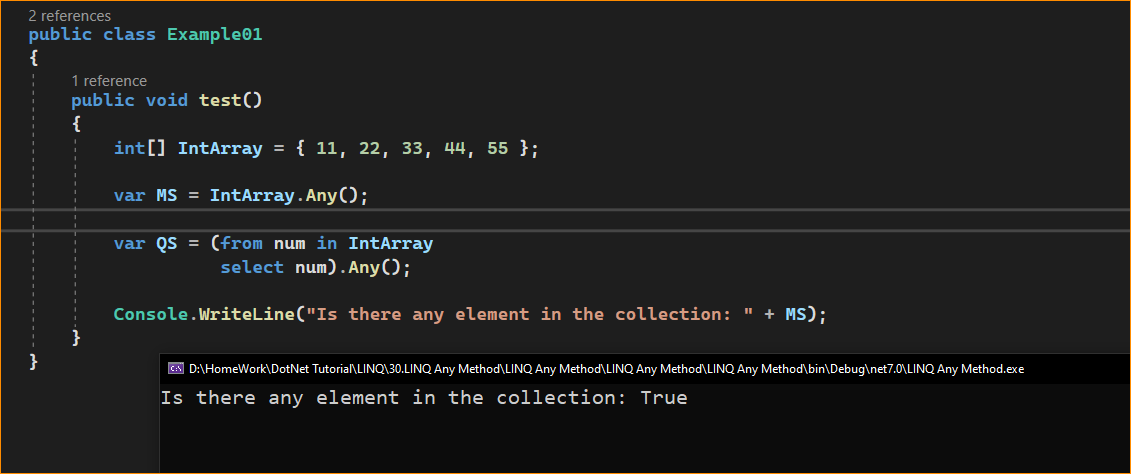
The LINQ Any Method in C# is used to check whether at least one of the elements of a data source satisfies a given condition or not. If any of the elements satisfy the given condition, then it returns true else returns false. It is also used to check whether a collection contains some element or not. That means it checks the length of the collection also. If it contains any data then it returns true else returns false. There are two overloaded versions of the Any Method method available in LINQ. They are as follows.



As you can see from the above image, the first overloaded version of the ANY Extension method does not take any parameter while the other overloaded version takes a predicate as a parameter. So, we need to use the First overloaded version to check whether the collection contains any element or not. We need to use the second overloaded version when we need to specify a predicate i.e. a condition.

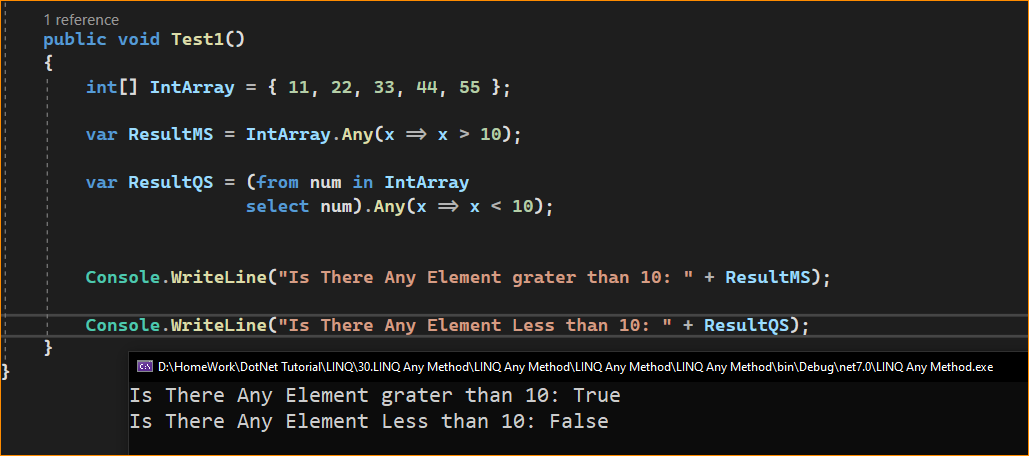
1. Example to Understand First Overloaded Version of LINQ Any Method in C#

Let us see an Example to Understand the First Overloaded Version of LINQ Any Method in C# which does not take any parameter using both Method and Query Syntax. As discussed, this method will return true if the collection on which it is applied contains at least one element else it will return false. For a better understanding, please have a look at the following example. The following example returns true as the collection contains at least one element. There is no such operator called “any” available in Query syntax, so we need to use the Mixed syntax.



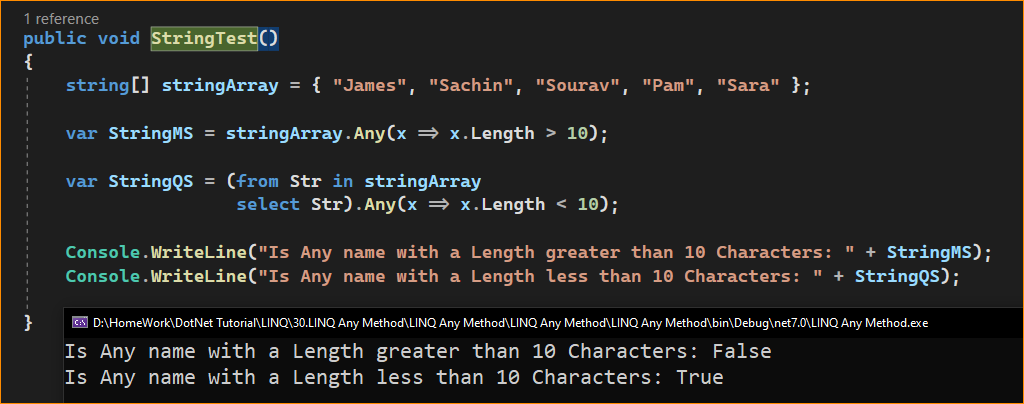
1. Example to Understand Second Overloaded Version of LINQ Any Method in C#

Let us see an Example to Understand the Second Overloaded Version of LINQ Any Method in C# which take a predicate as a parameter using both Method and Query Syntax. For a better understanding, please have a look at the following example. Our requirement is to check whether the collection contains at least one element which is less than 10. So, here, using the LINQ ALL method, using the predicate we need to specify the given condition i.e. the number is less than 10. The following example returns False as there is no element present in the intArray that is less than 10.



1. Example to Understand LINQ Any Method in C# using String Type

Let us see an example to Understand How to use LINQ Any Method in C# using String type collection. For a better understanding, please have a look at the following example which shows how to use Any Method in C# with String type collection. The following example will return True as some of the names are greater than 10 characters.

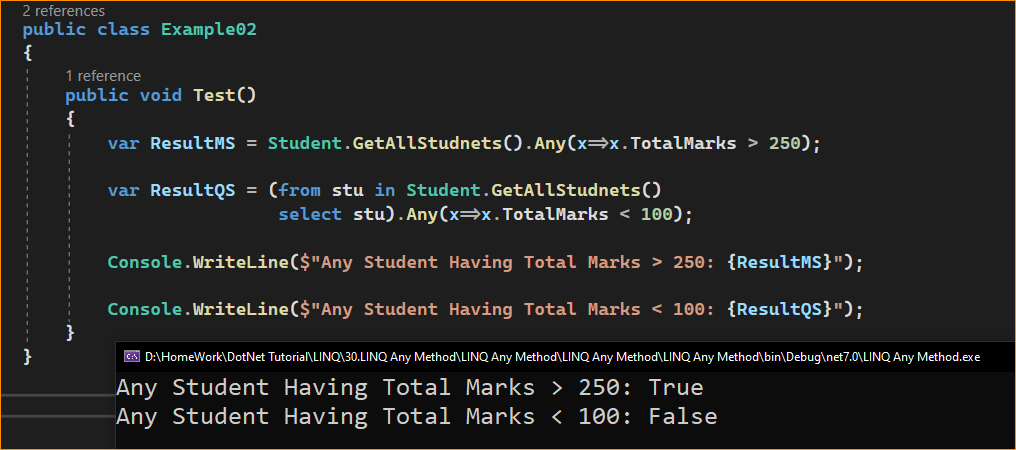


1. Example to Understand LINQ Any Method with Complex Type in C#:

Let us see an example to Understand How to use LINQ Any Method with Complex Data Type in C# using both Method and Query Syntax. We are going to work with the following Student and Subject classes. Create a class file with the name Student.cs and then copy and paste the following code into it. As you can see, the Student class has four properties i.e. ID, Name, TotalMarks, and Subjects. Here, within the Student class, we have also created one method i.e. GetAllStudnets() which will return the list of all the students. The Subject class has only two properties i.e. SubjectName and Marks.



Now, our requirement is to check whether any of the students have total marks greater than 250. As you can see except for the student James, all other students have a total mark greater than 250. So here, the LINQ Any method will give you the output as True. This is because the Any method will return true when any of the elements present in the collection satisfy the given condition.



1. Complex Example to Understand LINQ Any Method in C#:

Let us see a more Complex Example to Understand the LINQ Any Method in C#. If you see our collection, then you will observe that each student object has another collection called Subjects. Now we need to fetch all the student details whose mark on any subject is greater than 90. That means, now we will not apply the LINQ Any Extension method to the student’s collection, rather we will apply the LINQ Any method to the Subject property of the student object.

For a better understanding, please have a look at the following example. As we know, the Where Extension method takes a predicate as a parameter which will return a boolean true and false. Boolean TRUE means that the element will return and False means that the element will not return. Here, within the Where Extension method, on the Subject property (which is a collection) of the student object, we are applying the LINQ Any method. Now, for each student, the LINQ Any method will execute and it will check whether any of the Subject Marks satisfied the given condition i.e. Marks > 90, and if satisfied, the Any Method will return True, and Where extension method will return that Student object in output.

