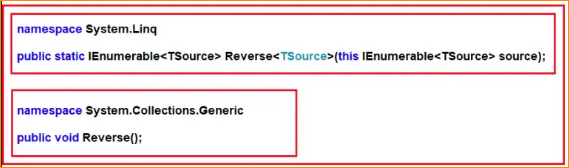
1. What is Reverse Method in C#?

The LINQ Reverse method is used to reverse the data stored in a data source. That means this method will not change the data rather it simply reverses the data stored in the data source. As a result, we will get the output in reverse order.

1. Reverse Method in C#:

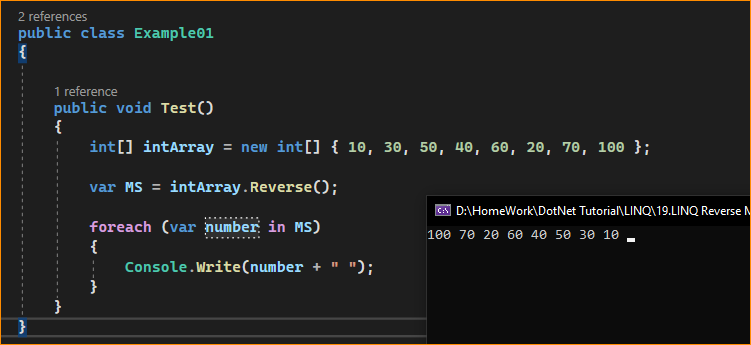
The Reverse Method in C# is implemented in two different namespaces. This method is implemented in System.LInq namespace as well as in System.Collections.Generic namespaces. Let us have a look at the signature or definition of the Reverse Method which is shown in the below image.

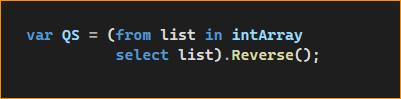


As you can see the Reverse Method which belongs to System.Linq namespace is implemented as an extension method on IEnumerable<TSource> interface and more importantly, this method also returns an IEnumerable<TSource> type. On the other hand, the Reverse method belongs to the System.Collections.Generic namespace is not returning any value as the return type is void. With this kept in mind, let us see some examples for a better understanding of the Reverse method in C#.

1. Example to Understand System.Linq Reverse Method in C#

Let us see an example to Understand the System.Linq namespace Reverse Method in C#. Please have a look at the below example, In the below example, we created one integer array to store integer numbers. And then we are calling the Reverse method on the intArray which will return a new collection of int type with data in reverse order. And then printing the data using a for each loop. In the below example, it is going to use the Reverse method which is defined inside the System.Linq namespace. Here, we are using the Method Syntax.

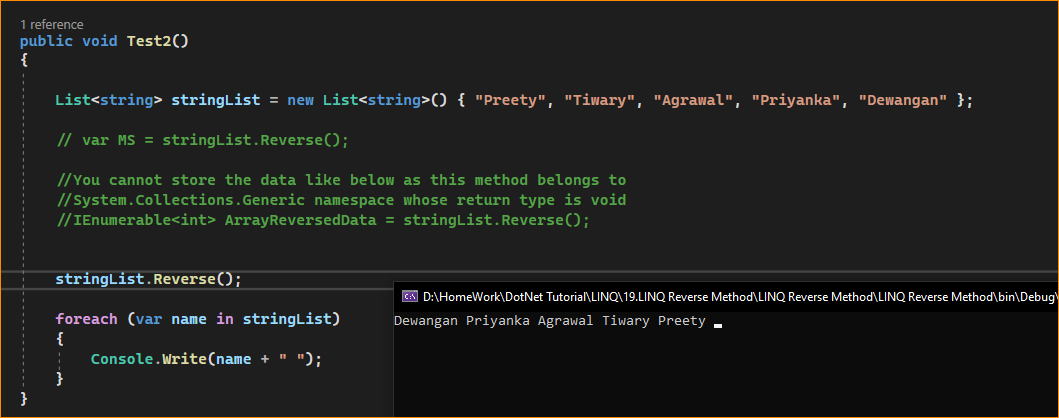




Note: In the above example, if you go to the definition of the Reverse method then you will see that this Reverse method belongs to the System.Linq namespace and hence we can able to store the data in a variable of type IEnumerable<int> as the source contains integer data.

1. Example to Understand System.Collections.Generic Reverse Method in C#

Let us see an example to Understand the System.Collections.Generic namespace Reverse Method in C#. In order to understand the System.Collections.Generic Reverse method, we need to create a collection of List<T> types. Please have a look at the below example, In the below example, we created one list collection to store string values. And then we are calling the Reverse method on the stringList collection which will reverse data in the original collection. And then printing the original collection data using a for each loop. In the below example, it is going to use the Reverse method which is defined inside the System.Collections.Generic namespace. As this method return type is void, so you can store the result in a variable as we did in our previous example.



1. How to Apply the LINQ Reverse Method on a Collection of List<T> Type in C#?

If you want to apply the Reverse method which belongs to System.Linq namespace on a collection of type List<T>, then first you need to convert the List<T> collection to IEnumerable or IQueryable by using the AsEnumerable() or AsQueryable() method on the data source.

If you use the AsEnumerable() method then it will convert the collection to IEnumerable whereas if you use AsQueryable() method then it will convert the collection to IQueryable. The following program shows how to apply the LINQ Reverse method on a collection of type List<T>.

