1. LINQ SkipWhile Method in C#

The LINQ SkipWhile Method in C# is used **to skip all the elements from a data source, a sequence, or a collection until a specified condition is true.** **Once the condition is failed, then returns the remaining element from the sequence as output.** The most important point you need to remember is that once the condition fails, the SkipWhile method will not check the rest of the elements even though the condition is true for some remaining elements.

The LINQ SkipWhile method belongs to the System.Linq namespace can be applied to any type implementing IEnumerable<T>. SkipWhile is useful in scenarios where you need to skip a contiguous portion of a sequence based on some condition and then process the rest of the elements. You need to remember that, like the Skip Method, the SkipWhile Method will not change the positions of the elements in the data source.

SkipWhile takes a predicate function that specifies the **condition to test each element**. It continues to skip elements in the sequence as long as this condition is true. **Once an element is encountered for which the condition is false**, SkipWhile yields that element and all subsequent elements. There are two overloaded versions available for this SkipWhile Method in C#. They are as shown in the below image.

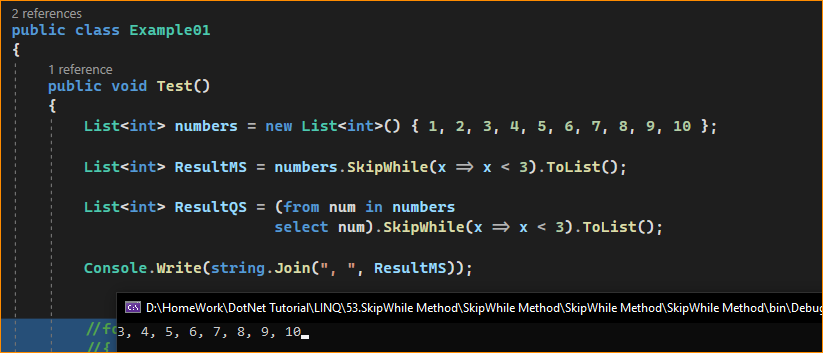


The First version of the LINQ SkipWhile method skips or bypasses the elements from a sequence as long as the specified condition is true. Once the condition fails, it will return the remaining elements from the data source.

The second overloaded version of the SkipWhile method bypasses or skips the elements from a sequence as long as the given condition is true and then returns the remaining elements. The second parameter of the function represents the index of the source elements. The element’s index can be used in the logic of the predicate function.

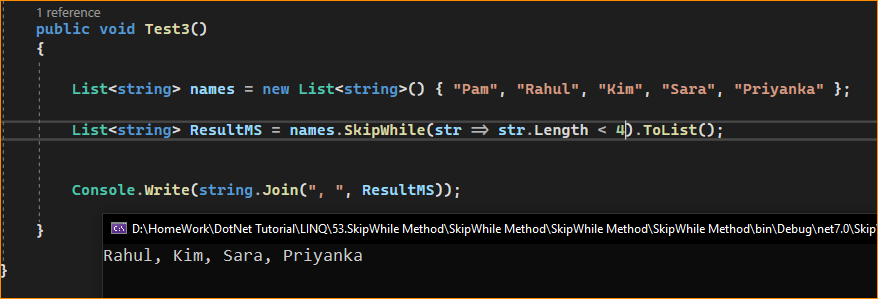
1. Example to Understand LINQ SkipWhile Method in C#:

Let us see an example of understanding the LINQ SkipWhile Method in C# using Method syntax and Query Syntax. In the following example, we created a data source that contains numbers from 1 to 10. Then, we skip the elements from the data source using the SkipWhile method with the condition that the number is less than 5. You need to remember that no such operator called skip is available in LINQ to write the query syntax. So, here, we need to use mixed syntax, as shown in the example below.



* Example:

In the following example, we have a collection of names. Here, we need to skip the names’ starting from the beginning whose length is less than 4.



Key Points

* Use Case: SkipWhile is used when you need to skip an unknown number of elements that meet a certain condition at the start of a sequence.
* Deferred Execution: Like many LINQ methods, SkipWhile uses deferred execution.
* Order-Dependent: The behavior of SkipWhile depends on the order of the elements in the sequence. It stops skipping when it finds an element that does not satisfy the condition.
* Difference from Where: Unlike Where, which filters out elements that don’t meet the condition throughout the entire sequence, SkipWhile only skips elements until it reaches an element that fails the condition, then yields all subsequent elements regardless of whether they meet the condition.



1. When to Use the LINQ SkipWhile Method in C#?

The LINQ SkipWhile method in C# is used in scenarios where you need to bypass a contiguous sequence portion based on a specified condition and then include the remaining elements. Here are some typical use cases:

* Processing Data After a Condition is Met: When you’re interested in processing elements of a sequence only after a certain condition is first false. For example, in a list of chronological events, you might skip events until a specific event occurs and then process all subsequent events.
* Ignoring Initial Subset of Data: In scenarios where the initial part of the data is irrelevant or needs to be excluded based on a condition. For instance, in financial calculations, you might want to skip initial transactions of a certain type and start aggregating data only after encountering a different type.
* Sequential Data Analysis: When working with ordered data, elements must be ignored until a particular criterion stops being true. For example, reading sensor data and skipping values until a certain threshold is crossed.
* File Processing: In file processing tasks, you might want to skip lines until a line with a certain property is encountered, then process the rest of the file.
* Data Transformation: In data transformation pipelines, you need to exclude the start of a data stream based on some dynamic condition before applying further transformations.
* Conditional Parsing: When parsing data, SkipWhile can be used to bypass parts of the data that are not needed or until the data meets a certain condition.

SkipWhile is useful in ordered sequences where the condition for skipping elements is contingent on the sequence’s order. It’s different from Where in that Where filters out elements throughout the sequence based on the condition. In contrast, SkipWhile stops skipping when an element fails to meet the condition and includes all subsequent elements.