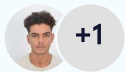




C# Concepts →

[Docs](#) / [C#](#) / [Deque](#)

Deque



Published Oct 2, 2024 • Updated Oct 18, 2024

[Contribute to Docs](#) →

Deque (Double-Ended Queue) is a type of data structure that allows insertion and removal of elements from both the front and rear. In `C#`, it can be implemented using `LinkedList<T>` and `List<T>`.

Creating a Deque using LinkedList Class

To create a deque in `C#`, use `LinkedList<T>`, where `T` defines the type of elements stored.

```
LinkedList<T> deque = new LinkedList<T>();
```

- `T`: Specifies the element type.

Example

The below example shows how to implement deque using `LinkedList`.

```
using System;
using System.Collections.Generic;

class Program {
```

```
static void Main() {  
    LinkedList<int> deque = new LinkedList<int>();  
  
    // Adding elements to the front  
    deque.AddFirst(1);  
    deque.AddFirst(2);  
  
    // Adding elements to the back  
    deque.AddLast(3);  
    deque.AddLast(4);  
  
    // Removing elements from the front  
    int front = deque.First.Value;  
    deque.RemoveFirst();  
  
    // Removing elements from the back  
    int back = deque.Last.Value;  
    deque.RemoveLast();  
  
    foreach (int value in deque) {  
        Console.WriteLine(value);  
    }  
}
```

Output:

```
1  
3
```

Creating a Deque using List Class

To create a deque in C#, use `List<T>`, where `T` defines the type of elements stored.

```
List<T> deque = new List<T>();
```

- `T`: Specifies the element type.

Example

The below example shows how to implement deque using `List`.

```
using System;
using System.Collections.Generic;

class Program {
    static void Main() {
        List<int> deque = new List<int>();

        // Adding elements to the front
        deque.Insert(0, 1);
        deque.Insert(0, 2);

        // Adding elements to the back
        deque.Add(3);
        deque.Add(4);

        // Removing elements from the front
        int front = deque[0];
        deque.RemoveAt(0);

        // Removing elements from the back
        int back = deque[deque.Count - 1];
        deque.RemoveAt(deque.Count - 1);


        foreach (int value in deque) {
            Console.WriteLine(value);
        }
    }
}
```

Output:

```
1
3
```

Codebyte Example

Use this example to experiment with implementing a `Deque` using `LinkedList`. Enjoy coding!



< Code	Output >
<pre>1 using System; 2 using System.Collections.Generic; 3 4 public class Example 5 { 6 public static void Main() 7 { 8 LinkedList<int> deque = new Lin 9 10 // Add elements to the front and 11 deque.AddLast(10); // Rear 12 deque.AddFirst(5); // Front 13 deque.AddLast(15); // Rear 14</pre>	

Run

All contributors



@anjar.bra



Anonymous contributor

Contribute to Docs

- [Learn more](#) about how to get involved.
- [Edit this page](#) on GitHub to fix an error or make an improvement.

- [Submit feedback](#) to let us know how we can improve Docs.

Learn C# on Codecademy

Career path

Computer Science

Looking for an introduction to the theory behind programming? Master Python while learning data structures, algorithms, and more!

Includes **6 Courses**

 With **Professional Certification**

 **Beginner Friendly**

75 hours

Free course

Learn C#

Learn Microsoft's popular C# programming language, used to make websites, mobile apps, video games, VR, and more.

 **Beginner Friendly**

23 hours

 [Back to top](#)