

Console Class

A console is an operating system window through which a user can communicate with the operating system or we can say a console is an application in which an input from the keyboard can be given and get the text as an output from the computer end. The command prompt is an example of a console in the windows and which accept MS-DOS commands. The console contains two attributes named as screen buffer and a console window.

In C#, the Console class is used to represent the standard input, output, and error streams for the console applications. This class is defined under System namespace. This class does not contain any constructor. Instead of the constructor, this class provides different types of properties and methods to perform operations.

Properties

PROPERTY	DESCRIPTION	Example
1. BackgroundColor	Gets or sets the background color of the console.	<pre>Console.WriteLine("Default Background Color: {0}", Console.BackgroundColor); // Set the Background color to blue Console.BackgroundColor = ConsoleColor.Blue; // Display current Background color Console.WriteLine("Changed Background Color: {0}", Console.BackgroundColor);</pre>
2. BufferHeight	Gets or sets the height of the buffer area.	<pre>Console.WriteLine("Default Buffer Height: {0}", Console.BufferHeight); // Set the Buffer Height to 100 Console.BufferHeight = 100; // Display current Buffer Height Console.WriteLine("Changed Buffer Height: {0}", Console.BufferHeight);</pre>
3. BufferWidth	Gets or sets the width of the buffer area.	<pre>Console.WriteLine("Default Buffer Width: {0}", Console.BufferWidth);</pre>

		<pre>// Set the Buffer Width to 100 Console.BufferWidth = 100; // Display current Buffer Width Console.WriteLine("Changed Buffer Width: {0}", Console.BufferWidth);</pre>
4. CapsLock	Gets a value indicating whether the CAPS LOCK keyboard toggle is turned on or turned off.	<pre>// Check if CAPS LOCK is on or off Console.WriteLine("Is CAPS LOCK on: {0}", Console.CapsLock);</pre>
5. CursorLeft	Gets or sets the column position of the cursor within the buffer area.	<pre>// Get the CursorLeft position Console.WriteLine("Current CursorLeft position: {0}", Console.CursorLeft); // Set the CursorLeft position Console.CursorLeft = 25; // Get the CursorLeft position Console.WriteLine("Current CursorLeft position: {0};", Console.CursorLeft);</pre>
6. CursorSize	Gets or sets the height of the cursor within a character cell.	<pre>// Get the CursorSize Console.WriteLine("Current CursorSize: {0}", Console.CursorSize); // Set the CursorSize Console.CursorSize = 100; // Get the CursorSize Console.WriteLine("Current CursorSize: {0}", Console.CursorSize);</pre>
7. CursorTop	Gets or sets the row position of the cursor within the buffer area.	<pre>// Get the CursorTop position Console.WriteLine("Current CursorTop position: {0}", Console.CursorTop); // Set the CursorTop position Console.CursorTop = 10;</pre>

		<pre>// Get the CursorTop position Console.Write("Current CursorTop position: {0};", Console.CursorTop);</pre>
8. CursorVisible	Gets or sets a value indicating whether the cursor is visible.	<pre>// Get the CursorVisible Console.WriteLine("Current Cursor Visible: {0}", Console.CursorVisible); // Set the CursorVisible Console.CursorVisible = false; // Get the CursorVisible Console.Write("Current Cursor Visible: {0}", Console.CursorVisible);</pre>
9. ForegroundColor	Gets or sets the foreground color of the console.	<pre>// Display current Foreground color Console.WriteLine("Default Foreground Color: {0}", Console.ForegroundColor); // Set the Foreground color to blue Console.ForegroundColor = ConsoleColor.Blue; // Display current Foreground color Console.WriteLine("Changed Foreground Color: {0}", Console.ForegroundColor);</pre>
10. LargestWindowHeight	Gets the largest possible number of console window rows, based on the current font and screen resolution.	<pre>// Display Largest Window Height Console.WriteLine("Largest Window Height: {0}", Console.LargestWindowHeight);</pre>
11. LargestWindowWidth	Gets the largest possible number of console window columns, based on the current font and	<pre>// Display Largest Window Width Console.WriteLine("Largest Window Width: {0}", Console.LargestWindowWidth);</pre>

	screen resolution.	
12. NumberLock	Gets a value indicating whether the NUM LOCK keyboard toggle is turned on or turned off.	<pre>// Check if NUM LOCK is on or off Console.WriteLine("Is NUM LOCK on: {0}", Console.NumberLock);</pre>
13. Out	Gets the standard output stream.	
14. OutputEncoding	Gets or sets the encoding the console uses to write output.	
15. Title	Gets or sets the title to display in the console title bar.	<pre>// Display current Title Console.WriteLine("Default Title: {0}", Console.Title); // Set the Title to GeeksForGeeks Console.Title = "GeeksForGeeks"; // Display current Title Console.WriteLine("Changed Title: {0}", Console.Title);</pre>
16. WindowHeight	Gets or sets the height of the console window area.	<pre>// Get the WindowHeight Console.WriteLine("Current WindowHeight: {0}", Console.WindowHeight); // Set the WindowHeight Console.WindowHeight = 50; // Get the WindowHeight Console.WriteLine("Current WindowHeight: {0}", Console.WindowHeight);</pre>
17. WindowLeft	Gets or sets the leftmost position of the console window area relative to the screen buffer.	<pre>// Get the WindowWidth Console.WriteLine("Current WindowLeft: {0}", Console.WindowLeft); // Set the WindowWidth Console.BufferWidth = 100;</pre>

		<pre> Console.WindowLeft = 10; // Get the WindowWidth Console.Write("Current WindowLeft: {0}", Console.WindowLeft); </pre>
18. WindowTop	Gets or sets the top position of the console window area relative to the screen buffer.	<pre> // Get the WindowWidth Console.WriteLine("Current WindowTop: {0}", Console.WindowTop); // Set the WindowWidth Console.BufferHeight = 100; Console.WindowTop = 10; // Get the WindowWidth Console.Write("Current WindowTop: {0}", Console.WindowTop); </pre>
19. WindowWidth	Gets or sets the width of the console window.	<pre> // Get the WindowWidth Console.WriteLine("Current WindowWidth: {0}", Console.WindowWidth); // Set the WindowWidth Console.WindowWidth = 150; // Get the WindowWidth Console.Write("Current WindowWidth: {0}", Console.WindowWidth); </pre>

Example:

```

// C# program to illustrate how to get
// Background and Foreground color
// of the console
using System;

```

```

public class GFG {

```

```

    static public void Main()
    {

```

```

// Get the Background and foreground
// color of Console Using BackgroundColor
// and ForegroundColor property of Console
Console.WriteLine("Background color :{0}",
    Console.BackgroundColor);

Console.WriteLine("Foreground color : {0}",
    Console.ForegroundColor);
}
}

```

Output:
Background color : Black
Foreground color : Black

Methods

METHOD	DESCRIPTION	Example
1. Beep()	Plays the sound of a beep through the console speaker.	<pre> int n = 5; // Play beep sound n times for (int i = 1; i < n; i++) Console.Beep(); </pre>
2. Clear()	Clears the console buffer and corresponding console window of display information.	<pre> // Print the statements Console.WriteLine("GeeksForGeeks"); Console.WriteLine("A Computer Science Portal"); Console.WriteLine("For Geeks"); // Clear the Console Console.Clear(); </pre>
3. MoveBufferArea()	Copies a specified source area of the screen buffer to a specified destination area.	<p>Parameters:</p> <pre> sourceLeft: The leftmost column of the source area. sourceTop: The topmost row of the source area. </pre>

		<p>sourceWidth: The number of columns in the source area.</p> <p>sourceHeight: The number of rows in the source area.</p> <p>targetLeft: The leftmost column of the destination area.</p> <p>targetTop: The topmost row of the destination area.</p> <pre> Console.WriteLine("GeeksForGeeks"); // using the method Console.MoveBufferArea(0, 0, Console.BufferWidth, Console.BufferHeight, 10, 10); </pre>
4. Read()	Reads the next character from the standard input stream.	<pre> int x; Console.WriteLine("Enter your Character to get Decimal number"); // using the method x = Console.Read(); Console.WriteLine(x); </pre>
5. ReadKey()	Obtains the next character or function key pressed by the user. The pressed key is displayed in the console window.	<pre> Console.ReadKey(); </pre>
6. ReadLine()	Reads the next line of characters from the standard input stream.	<pre> name = Console.ReadLine(); </pre>
7. ResetColor()	Sets the foreground and background console colors to their defaults.	<pre> // using BackgroundColor property Console.BackgroundColor = ConsoleColor.Yellow; // using ForegroundColor property Console.ForegroundColor = ConsoleColor.Red; </pre>

		<pre> Console.WriteLine("Welcome to GeeksForGeeks"); // using ResetColor() Method Console.ResetColor(); Console.WriteLine("Welcome to GeeksForGeeks"); </pre>
8. <code>SetBufferSize(Int32, Int32)</code>	Sets the height and width of the screen buffer area to the specified values.	<pre> Console.SetBufferSize(800, 800); Console.WriteLine("Start"); while (true) { Console.WriteLine("Great Geek's Example!!!"); } </pre>
9. <code>SetCursorPosition(Int32, Int32)</code>	Sets the position of the cursor.	<pre> Console.SetWindowSize(40, 40); // setting buffer size of console Console.SetBufferSize(80, 80); // using the method Console.SetCursorPosition(20, 20); Console.WriteLine("Hello GFG!"); Console.Write("Press any key to continue . . . "); Console.ReadKey(true); </pre>
10. <code>SetWindowPosition(Int32, Int32)</code>	Sets the position of the console window relative to the screen buffer.	<pre> Console.SetWindowSize(20, 20); // setting buffer size Console.SetBufferSize(80, 80); // using the method Console.SetWindowPosition(0, 0); Console.WriteLine("Hello GFG!"); Console.Write("Press any key to continue . . . "); Console.ReadKey(true); </pre>
11. <code>SetWindowSize(Int32, Int32)</code>	Sets the height and width of the console	<pre> // Passed 40, 40 to SetWindowSize to </pre>

	<p>window to the specified values.</p>	<pre>// change window size to 40 by 40 Console.SetWindowSize(40, 40); // Printing the current dimensions Console.WriteLine(Console.WindowWidth); Console.WriteLine(Console.WindowHeight);</pre>
--	--	--