

C# – Read User Input through Console

C# – Read From Console

In C#, you can read input from user through console using `Console.ReadLine()` function. `Console.ReadLine()` reads the input entered by user until enter is pressed, and the value is returned.

We can store the returned value in a variable and access it.

In the following sections, we will learn how to read a string and a number from console with the help of examples.

C# – How to Read String From Console

By default `Console.ReadLine()` reads string, from the console, entered by user. We are storing the value into a string and printing it in the next statement.

Program.cs

```
using System;

namespace CSharpExamples {

    class Program {
        static void Main(string[] args) {
            Console.Write("Enter string: ");
            String str = Console.ReadLine();
            Console.WriteLine("You entered \""+str+"\" in the console.");
        }
    }
}
```

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run
Enter string: TutorialKart
You entered "TutorialKart" in the console.
```

C# – How to Read a Number From Console

We can read a number from the console using `Console.ReadLine()`. The trick here is, read the input from user

We can read a number from the console using `Console.ReadLine()`. The trick here is, read the input from user as a string and convert it to integer.

Program.cs

```
using System;

namespace CSharpExamples {

    class Program {
        static void Main(string[] args) {
            Console.Write("Enter number: ");
            int n = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("You entered the number, "+n+" in console.");
        }
    }
}
```

Output

```
PS D:\workspace\csharp\HelloWorld> dotnet run
Enter number: 65
You entered the number, 65 in console.
```

Conclusion

In this [C# Tutorial](#), we read input entered by the user in console using `Console.ReadLine()` function.

C# Tutorial

- ◆ [C# Tutorial](#)
- ◆ [C# Basic Example](#)
- ◆ [C# Comments](#)
- ◆ [C# Variables](#)
- ◆ [C# Constants](#)
- ◆ [C# if, if-else](#)
- ◆ [C# switch](#)
- ◆ [C# while loop](#)
- ◆ [C# for loop](#)
- ◆ [C# foreach](#)
- ◆ [C# break](#)

- ◆ C# continue
- ◆ C# struct
- ◆ C# enum
- ◆ C# String
- ◆ C# Array
- ◆ C# Command Line Arguments

C# Console Operations

- ◆ C# Write to Console
- ⇒ C# Read from Console

C# Object Oriented Programming Concepts

- ◆ C# Class & Constructors
- ◆ C# Encapsulation
- ◆ C# Polymorphism
- ◆ C# Method Overloading
- ◆ C# Interfaces

C# String Operations

- ◆ C# String Length
- ◆ C# Substring

C# File Operations

- ◆ C# Read Text File
- ◆ C# Write to File
- ◆ C# Delete File
- ◆ C# Copy File

C# Exception Handling

- ◆ C# try-catch
- ◆ C# finally
- ◆ C# throw
- ◆ C# Custom Exception
- ◆ C# SystemException

◆ C# DivideByZeroException

◆ C# NullReferenceException

◆ C# InvalidCastException

◆ C# IOException

◆ C# FieldAccessException

C# Collections

◆ C# List

◆ C# SortedList

◆ C# HashSet

◆ C# SortedSet

◆ C# Stack

◆ C# Queue

◆ C# LinkedList

◆ C# Dictionary

◆ C# SortedDictionary

C# Errors [Solved]

◆ C# Error: Class does not contain a constructor that takes arguments