Source : C# Corner (www.c-sharpcorner.com) Print

ARTICLE

var Type:

- C# 3.0 adds the interesting behavior of Declaring Local Variables Implicitly. This means that there is no need to mention the data type when declaring a variable. A local variable can be declared implicitly using the **var** keyword in C#.
- Declaring local variables implicitly has some restrictions; the variable must be initialized to some expression that can not be null.

```
var a= 10;
var z = "Rekha";
```

- The primary reason for its existence is the introduction of anonymous types in C#
- Another point to stress is that variable inference does not work for class level fields or method arguments or anywhere else other than for local variables in a method.

Advantages :

- Less typing with no loss of functionality
- Increases the type safety of your code. A foreach loop using an iteration variable which is typed to var will catch silently casts that are introduced with explicit types
- Makes it so you don't have to write the same name twice in a variable declaration.
- Some features, such as declaring a strongly typed anonymous type local variable, require the use of var

Example 1:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;


namespace Namesp1
{
   class Program
   {
      static void Main(string[] args)
      {
         var x = Convert.ToInt32(Console.ReadLine());
         var y = Convert.ToInt32(Console.ReadLine());
         Console.WriteLine("The Sum Is : " + (x + y));
      }
```

```
    }
}
```

Example 2:

```
using System;
namespace ConsoleApplication1
{
   class A
   {
      public static void Main()
      {
         int sum = 0;
         int[] arr = new int[10];
         for (int i = 0; i < 10; i++)
         {
            arr[i] = i;
         }
         foreach (var x in arr)
         {
            Console.WriteLine("Value is: " + x);
            sum =sum + x;
         }
         Console.WriteLine("Sum of array elements is : " + sum);
      }
   }
}
```

Output:

```
Value is: 1
Value is: 2
Value is: 3
Value is: 4
Value is: 5
Value is: 6
Value is: 7
Value is: 8
Value is: 9
Value is: 10
Sum of array elements is : 55
```

Thank you for using C# Corner