

# C# Interactive in Visual Studio



byAnoop Kumar Sharma

CORE · Sep. 14, 18 · Web Dev Zone · Tutorial

Like (5)   Comment (0)   Save   Tweet

In this article, we will learn how to use C# Interactive in Visual Studio. The C# Interactive window has been a part of Visual Studio since the Visual Studio 2015 update 1 release. In this article, I am using Visual Studio 2017 for demonstration purposes. C# Interactive is a **REPL** Editor, i.e. **Read-Evaluate-Print-Loop** with an advanced editor. With the C# Interactive window, we can test our code snippet without compiling or running the complete code.

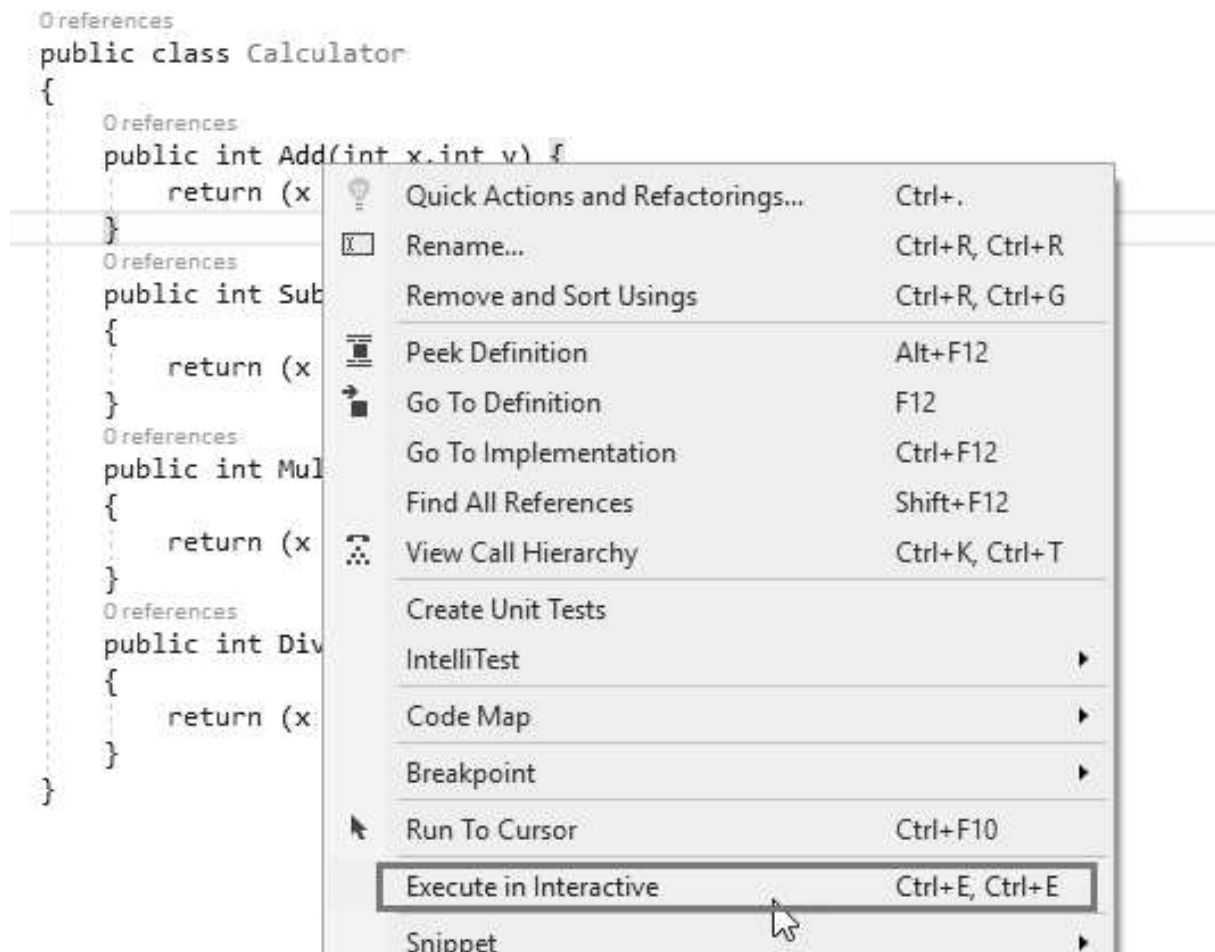
## Let's Begin:

Open C# Interactive window in Visual Studio. We can open C# Interactive window in multiple ways:

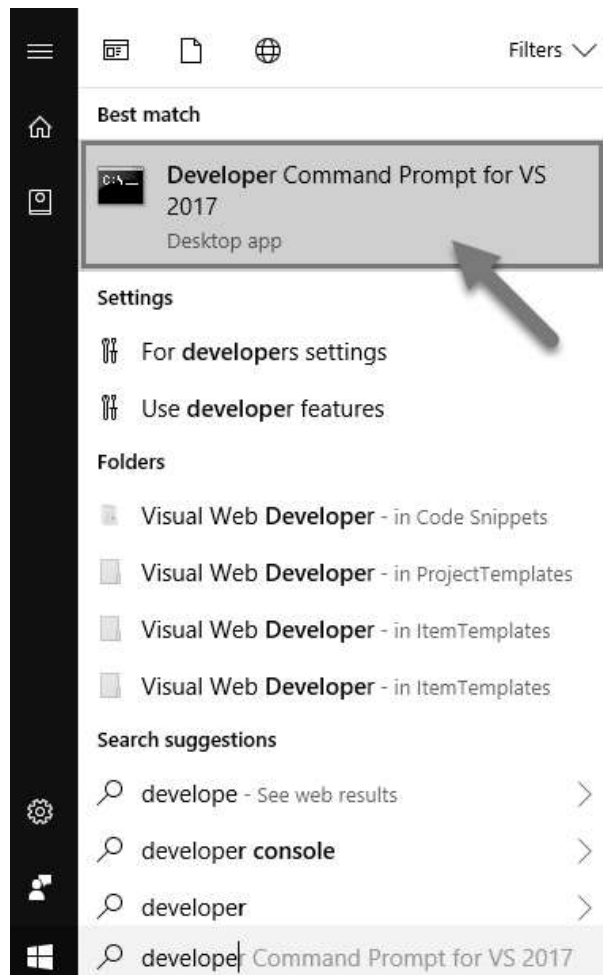
- From Menu Bar: Go to View Menu then move the cursor to Other Windows and then click on C# Interactive.



- From context box: Right click on the .cs file and then click on Execute in Interactive (shortcut is Ctrl+E, Ctrl+E).



- From Visual Studio command prompt: Open Visual Studio Developer command prompt.



Then type **csi** and press enter. C# Interactive mode will be available in the command prompt.

```
Developer Command Prompt for VS 2017 - csi

*****
** Visual Studio 2017 Developer Command Prompt v15.6.4
** Copyright (c) 2017 Microsoft Corporation
*****

C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise>csi
Microsoft (R) Visual C# Interactive Compiler version 2.7.0.62715
Copyright (C) Microsoft Corporation. All rights reserved.

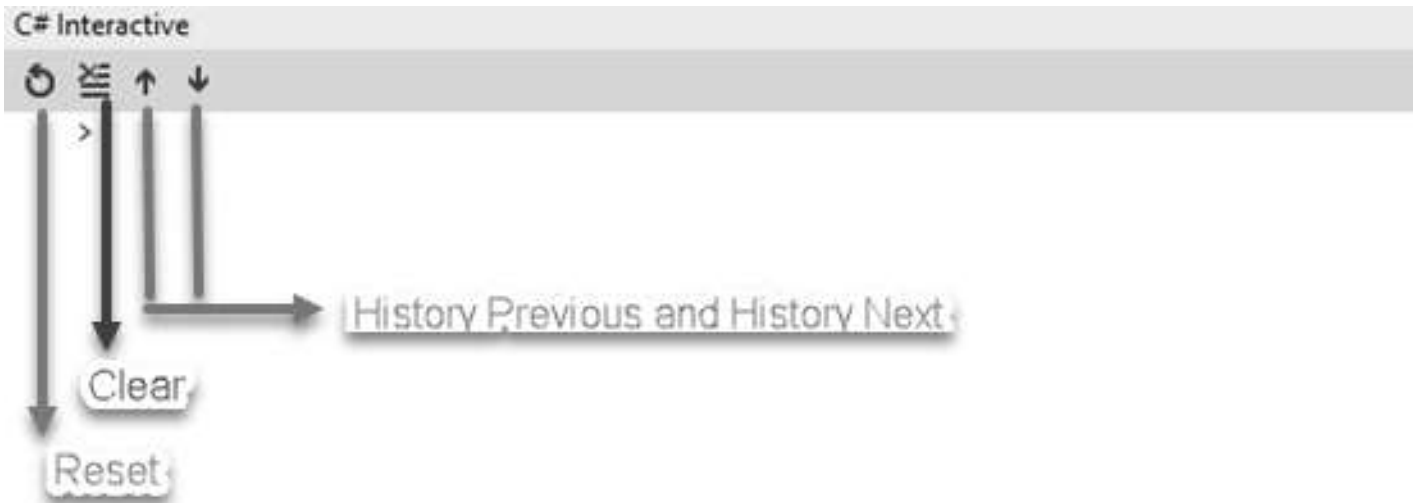
Type "#help" for more information.
> _
```

The C# Interactive window has 4 options in the window that is as below:

1. **Reset:** Reset will reset all the DLLs, methods, etc., loaded in the C# Interactive window.
2. **Clear:** Clear will clear the screen of the C# Interactive Window.

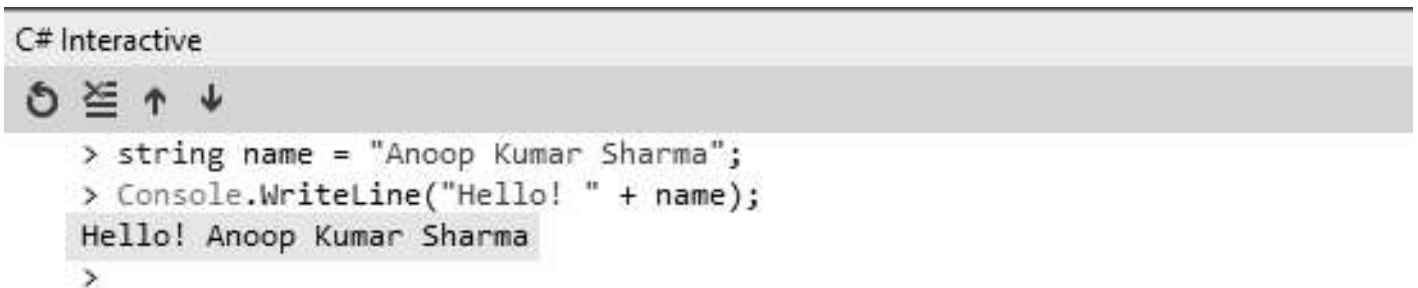
3. **History Previous:** For checking the previous step/command executed in the C# Interactive Window.

4. **History Next:** For checking the next step/command executed in the C# Interactive Window.



## Execute Code in C# Interactive Window

We can execute C# code in the interactive window easily. In the below example, I have declared a string variable and displayed the output in the C# Interactive Window. C# Interactive window supports C# 6 and C# 7 as well.



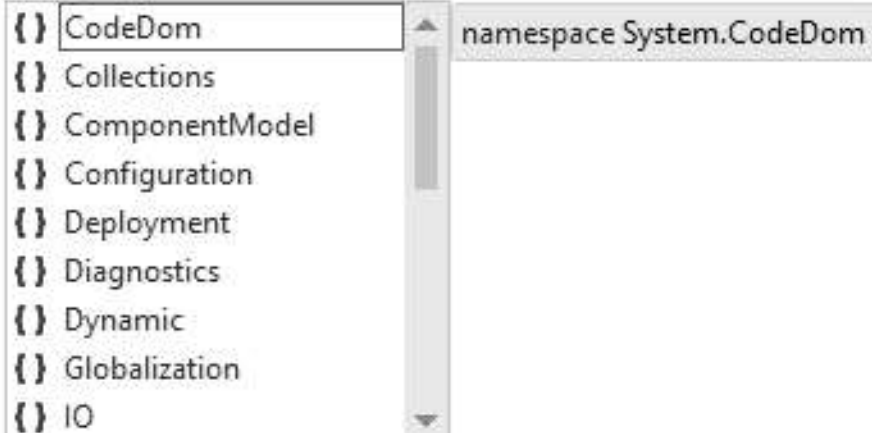
Intelligence feature is also available in the Interactive window, which makes it easy to use.

## C# Interactive



```
> using System.
```

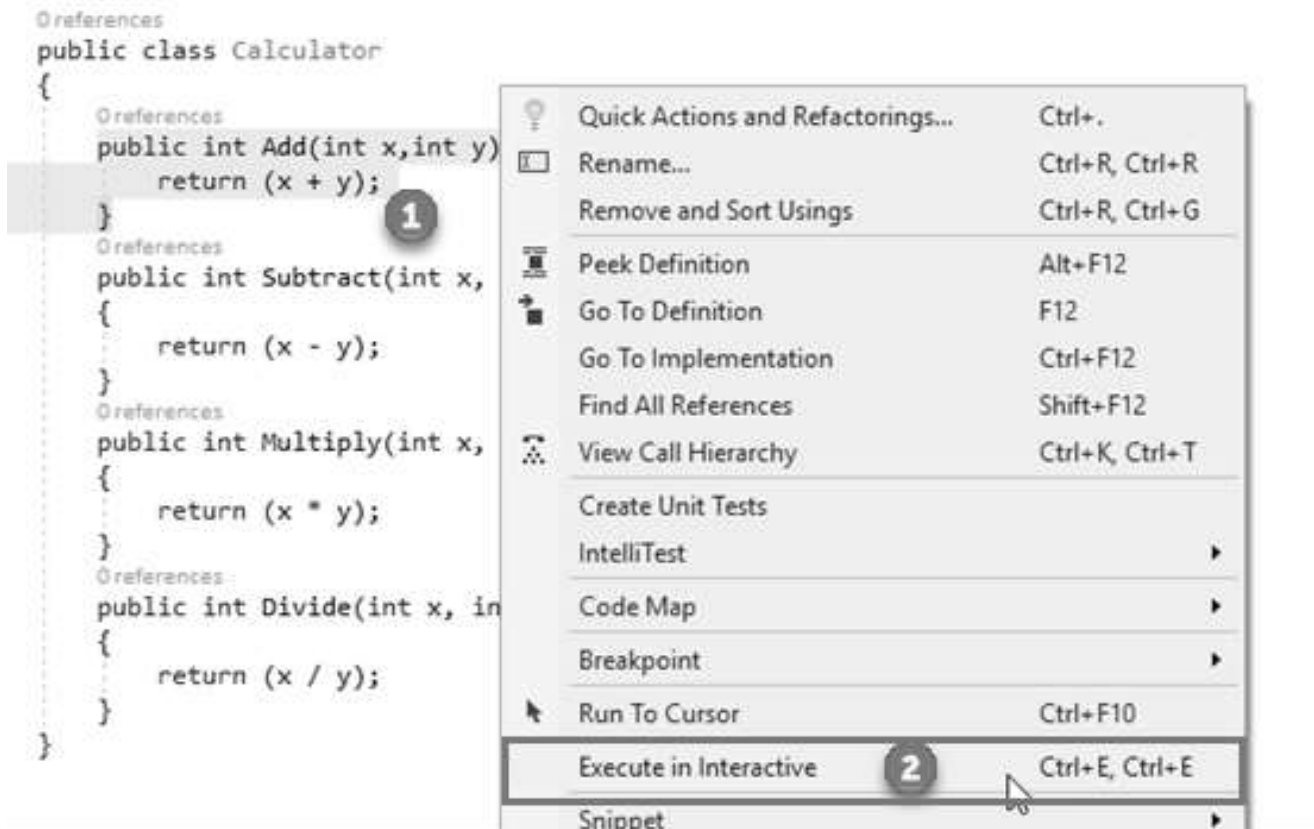
```
|
```



We can execute any code block in the Interactive Window. For this, we have to just select the method which we want to test/access in the C# Interactive Window. Check the below example, I have created a Calculator class which has an Add, Subtract, Multiply, and Divide method.

```
namespace CalcBL
{
    References
    public class Calculator
    {
        References
        public int Add(int x,int y) {
            return (x + y);
        }
        References
        public int Subtract(int x, int y)
        {
            return (x - y);
        }
        References
        public int Multiply(int x, int y)
        {
            return (x * y);
        }
        References
        public int Divide(int x, int y)
        {
            return (x / y);
        }
    }
}
```

Now select the method or function you want to test in the C# Interactive window and right-click and select Execute in Interactive.



Test the method by passing the parameter value in the method.

```

C# Interactive
> public int Add(int x,int y) {
.     return (x + y);
. }
> Console.WriteLine(Add(10, 50));
60
>

```

We can also load the DLL in REPL with the help of the `#r` command, followed by the path of the DLL to load in the C# Interactive Window.

```

C# Interactive
> #r "C:\Users\Anoop Kumar Sharma\source\repos\ConsoleCalculator\CalcBL\bin\Debug\CalcBL.dll"
> using CalcBL;
> Calculator objCalculator = new Calculator();
> Console.WriteLine(objCalculator.Multiply(10, 2));
20
>

```

↑  
Add DLL Reference in C# Interactive

C# Interactive Window also supports error handling in both compile time as well as runtime. Check the below screenshot:

```
C# Interactive
> Console.WriteLine(10 / 0);
(1,15): error CS0020: Division by constant zero
>
```

Hope this will help you. Thanks!