

# C# Example: Hello World

In C# programming language, a simple "hello world" program can be written by multiple ways. Let's see the top 4 ways to create a simple C# example:

- Simple Example
- Using System
- Using public modifier
- Using namespace

## C# Simple Example

```
class Program
{
    static void Main(string[] args)
    {
        System.Console.WriteLine("Hello World!");
    }
}
```

Output:

```
Hello World!
```

## Description

**class:** is a keyword which is used to define class.

**Program:** is the class name. A class is a blueprint or template from which objects are created. It can have data members and methods. Here, it has only Main method.

**static:** is a keyword which means object is not required to access static members. So it saves memory.

**void:** is the return type of the method. It doesn't return any value. In such case, return statement is not required.

**Main:** is the method name. It is the entry point for any C# program. Whenever we run the C# program, Main() method is invoked first before any other method. It represents start up of the program.

**string[] args:** is used for command line arguments in C#. While running the C# program, we can pass values. These values are known as arguments which we can use in the program.

**System.Console.WriteLine("Hello World!"):** Here, System is the namespace. Console is the class defined in System namespace. The WriteLine() is the static method of Console class which is used to write the text on the console.

## C# Example: Using System

If we write *using System* before the class, it means we don't need to specify System namespace for accessing any class of this namespace. Here, we are using Console class without specifying System.Console.

```
using System;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello World!");
    }
}
```

Output:

```
Hello World!
```

## C# Example: Using public modifier

We can also specify *public* modifier before class and Main() method. Now, it can be accessed from outside the class also.

```
using System;

public class Program
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Hello World!");
    }
}
```

Output:

```
Hello World!
```

# C# Example: Using namespace

We can create classes inside the namespace. It is used to group related classes. It is used to categorize classes so that it can be easy to maintain.

```
using System;
namespace ConsoleApplication1
{
    public class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Output:

```
Hello World!
```