

[< Previous](#)[Next >](#)

C# - Action Delegate

Action is a delegate type defined in the `System` namespace. An Action type delegate is the same as [Func delegate](#) except that the Action delegate doesn't return a value. In other words, an Action delegate can be used with a method that has a void return type.

For example, the following delegate prints an int value.

Example: C# Delegate

```
public delegate void Print(int val);

static void ConsolePrint(int i)
{
    Console.WriteLine(i);
}

static void Main(string[] args)
{
    Print prnt = ConsolePrint;
    prnt(10);
}
```

Output:

```
10
```

You can use an Action delegate instead of defining the above Print delegate, for example:

Example: Action delegate

```
static void ConsolePrint(int i)
{
    Console.WriteLine(i);
}

static void Main(string[] args)
{
    Action<int> printActionDel = ConsolePrint;
    printActionDel(10);
}
```

Try it

You can initialize an Action delegate using the new keyword or by directly assigning a method:

```
Action<int> printActionDel = ConsolePrint;

//Or

Action<int> printActionDel = new Action<int>(ConsolePrint);
```

An Action delegate can take up to 16 input parameters of different types.

An Anonymous method can also be assigned to an Action delegate, for example:

Example: Anonymous method with Action delegate

```
static void Main(string[] args)
{
    Action<int> printActionDel = delegate(int i)
    {
        Console.WriteLine(i);
    };

    printActionDel(10);
}
```

Try it

Output:

10

A Lambda expression also can be used with an Action delegate:

Example: Lambda expression with Action delegate

```
static void Main(string[] args)
{

    Action<int> printActionDel = i => Console.WriteLine(i);

    printActionDel(10);
}
```

Try it

Thus, you can use any method that doesn't return a value with Action delegate types.

Advantages of Action and Func Delegates

- 1. Easy and quick to define delegates.
- 2. Makes code short.
- 3. Compatible type throughout the application.



Points to Remember :

- 1) Action delegate is same as func delegate except that it does not return anything. Return type must be void.
- 2) Action delegate can have 0 to 16 input parameters.
- 3) Action delegate can be used with [anonymous methods](#) or [lambda expressions](#).

Learn C# using coding questions with answers and explanations.

C# Questions & Answers

Want to check how much you know C#?

Start C# Skill Test

Related Articles

- > [Difference between Array and ArrayList](#)
- > [Difference between Hashtable and Dictionary](#)
- > [How to write file using StreamWriter in C#?](#)
- > [How to sort the generic SortedList in the descending order?](#)
- > [Difference between delegates and events in C#](#)
- > [How to read file using StreamReader in C#?](#)
- > [How to calculate the code execution time in C#?](#)
- > [Design Principle vs Design Pattern](#)
- > [How to convert string to int in C#?](#)
- > [Boxing and Unboxing in C#](#)
- > [More C# articles](#)

 Share

 Tweet

 Share

 Whatsapp

< Previous

Next >

TUTORIALSTEACHER.COM

TutorialsTeacher.com is optimized for learning web technologies step by step. Examples might be simplified to improve reading and basic understanding. While using this site, you agree to have read and accepted our terms of use and privacy policy.

✉ feedback@tutorialsteacher.com

E-MAIL LIST

Subscribe to TutorialsTeacher email list and get latest updates, tips & tricks on C#, .Net, JavaScript, jQuery, AngularJS, Node.js to your inbox.

Email address

GO

We respect your privacy.

TUTORIALS

- > ASP.NET Core
 - > ASP.NET MVC
 - > IoC
 - > Web API
 - > C#
 - > LINQ
 - > Entity Framework
- > AngularJS 1
 - > Node.js
 - > D3.js
 - > JavaScript
 - > jQuery
 - > Sass
 - > Https