ref (C# Reference)

The ref keyword indicates a value that is passed by reference. It is used in three different contexts:

* In a method signature and in a method call, to pass an argument to a method by reference. See [Passing an argument by reference](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/ref#passing-an-argument-by-reference) for more information.
* In a method signature, to return a value to the caller by reference. See [Reference return values](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/ref#reference-return-values) for more information.
* In a member body, to indicate that a reference return value is stored locally as a reference that the caller intends to modify. See [Ref locals](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/ref#ref-locals) for more information.

# out (C# Reference)

You can use the out contextual keyword in two contexts:

* As a [parameter modifier](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/out-parameter-modifier), which lets you pass an argument to a method by reference rather than by value.
* In [generic type parameter declarations](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/out-generic-modifier) for interfaces and delegates, which specifies that a type parameter is covariant.

# Passing Parameters (C# Programming Guide)

In C#, arguments can be passed to parameters either by value or by reference. Passing by reference enables function members, methods, properties, indexers, operators, and constructors to change the value of the parameters and have that change persist in the calling environment. To pass a parameter by reference, use the ref or out keyword. For simplicity, only the ref keyword is used in the examples in this topic. For more information about the difference between ref and out, see [ref](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/ref), [out](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/out), and [Passing Arrays Using ref and out](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/passing-arrays-using-ref-and-out).

**Where to use Ref**: when we are calling a procedure with an in parameter and the same parameter will be used to store the output of that proc.

**Where to use Out:** when we are calling a procedure with no in parameter and teh same param will be used to return the value from that proc. Also note the output