Arrays are a fixed size collection of a homogeneous type.

Array size must be declared at initialisation. Once the size is set it cannot be changed.

Arrays can hold value or ref types.

string[] Colours; – basic array declaration

Colours = new string[4]; - initialise array.

var Colours = new string[4] – works at both declaration and initialisation. Developers prefer the var option.

Arrays are memory and performance efficient especially with large collections.

Do:

Use array when the size can be determined at initialisation.

Use a plural name.

Use Collection Initialisers

Avoid:

When the size of the collection is not known. E.G if data comes from a database

Manually populating an array

To simplify the process of populating an array we can use a technique called ‘collection initialisers’

e.g. string[] Colours = new string[4] { “”, “”, “”, “”};

further simplified to: string[] Colours = { “”, “”, “”, “”};

The iteration variable of a foreach loop is read-only.

$”the var {var}” – this technique is called string interpolation and is present in C#6.0

Foreach will iterate through every element in the index order

For loop can modify the elements of an array

Array derives from System.Array class. It provides static methods.

Array.IndexOf(theArray, theElement);

theArray.SetValue( index, theValue);

FAQ

1. When is it appropriate to use an array?

When the size of the collection of data is known beforehand and you’re dealing with a large amount of data for performance. Also when dealing with multidimensional data sets.

1. What’s the difference between a foreach and for loop?

Foreach loop allows iteration through every element of the array treating each element as a read-only value. For loop allows iteration control through arrays with more flexibility.