Cheatsheets / **Intermediate C#**

Learn C#: Arrays

C# Arrays

In C#, an *array* is a structure representing a fixed length ordered collection of values or objects with the same type.

Arrays make it easier to organize and operate on large amounts of data. For example, rather than creating 100 integer variables, you can just create one array that stores all those integers!

```
// `numbers` array that stores integers
int[] numbers = { 3, 14, 59 };

// 'characters' array that stores strings
string[] characters = new string[] { "Huey", "Dewey", "Louie" };
```

Declaring Arrays

A C# array variable is declared similarly to a non-array variable, with the addition of square brackets ([]) after the type specifier to denote it as an array.

The `new` keyword is needed when instantiating a new array to assign to the variable, as well as the array length in the square brackets. The array can also be instantiated with values using curly braces ({}). In this case the array length is not necessary.

```
// Declare an array of length 8 without setting the values.
string[] stringArray = new string[8];

// Declare array and set its values to 3, 4, 5.
int[] intArray = new int[] { 3, 4, 5 };
```

Declare and Initialize array

In C#, one way an array can be declared and initialized at the same time is by assigning the newly declared array to a comma separated list of the values surrounded by curly braces ({}). Note how we can omit the type signature and `new` keyword on the right side of the assignment using this syntax. This is only possible during the array’s declaration.

```
// `numbers` and `animals` are both declared and initialized with values.
int[] numbers = { 1, 3, -10, 5, 8 };
string[] animals = { "shark", "bear", "dog", "raccoon" };
```

Array Element Access

In C#, the elements of an array are labeled incrementally, starting at 0 for the first element. For example, the 3rd element of an array would be indexed at 2, and the 6th element of an array would be indexed at 5.
A specific element can be accessed by using the square bracket operator, surrounding the index with square brackets. Once accessed, the element can be used in an expression, or modified like a regular variable.

```
// Initialize an array with 6 values.
int[] numbers = { 3, 14, 59, 26, 53, 0 };

// Assign the last element, the 6th number in the array (currently 0), to 58.
numbers[5] = 58;

// Store the first element, 3, in the variable `first`.
int first = numbers[0];
```

C# Array Length

The *Length* property of a C# array can be used to get the number of elements in a particular array.

```
int[] someArray = { 3, 4, 1, 6 };
Console.WriteLine(someArray.Length); // Prints 4

string[] otherArray = { "foo", "bar", "baz" };
Console.WriteLine(otherArray.Length); // Prints 3
```

Print Share ▼