# Documentation Hunt

In addition to `Sort()`, `IndexOf()`, and `Find()`, there are several other built-in methods for arrays. You can find them (and you probably guessed it) in the [Microsoft documentation](#).

Let’s put your documentation sleuthing to use and hunt down some built-in methods to write some code!

☑**Instructions**

**1.**

In the [Microsoft documentation](#), find the method that allows you to copy your playlist to a new playlist called `summerStrutCopy`. Print the first value of `summerStrutCopy` to the playlist to see if is the same as `summerStrut`.

Hint

`Array.Copy()` ([documentation](#)) copies a range of elements from one array to a second array. It takes three parameters: the name of the array to be copied, the new array, and the length of the array elements.

```
string[] players = { "Emily", "Kyle", "Todd", "Rachel", "Grayson" };

// This creates a new array with default values
string[] playersCopy = new string[5];

// This will populate the playersCopy array with { "Grayson", "Rachel",
"Todd", "Kyle", "Emily" }
Array.Copy(players, playersCopy, 5);
```

**2.**
In the Microsoft documentation, find the method that reverses the order of the array elements. Use it to reverse the order of the `summerStrut` playlist. Check to see if it worked by printing the first and last songs to the console.

Hint

`Array.Reverse()` (documentation) will switch the order of elements in an entire array. It can also reverse them in a portion of an array, if the overload is used:

```
string[] players = { "Emily", "Kyle", "Todd", "Rachel", "Grayson" };

// This will return { "Grayson", "Rachel", "Todd", "Kyle", "Emily" }
Array.Reverse(players);
```

**3.**

In the Microsoft documentation, find the method that turns every rating in the `ratings` array to zero. Check to see if it worked by printing out the first value to the console (it should be to 0).

Hint

`Array.Clear()` (documentation) sets a range of elements in an array to the default value. It takes three parameters: the name of the array, the starting index of the range to clear, and the number of elements to clear.

To clear an entire array, set the index to 0 (if it is zero-indexed) and then pass in the length of the array for the third parameter.

```
string[] players = { "Emily", "Kyle", "Todd", "Rachel", "Grayson" };

// This will return { null, null, null, null, null }
Array.Clear(players, 0, players.Length);
```