Arrays

Imagine that you want to create a game with 100 enemies, and then you have to move each one of those enemies. With the knowledge you currently have, you would have to code and write a line for each enemy, which is not an expandable solution. There is a better solution: Arrays.

An array is a structure with a fixed length that contains several elements of the same type.

> **Example**
>
> An array of **strings** that contains all the best pizza toppings
> `string[] pizzaToppings = {"Pepperoni", "Sausage", "Mushrooms", "Bacon", "Onions", "Extra Cheese", "Peppers", "Chicken"}`

When you have an array, you can use logic to go through each item, saving a lot of coding.

Creating Arrays

Arrays are reference objects, and then the formula to create them is:

`type[] _arrayName;`

> **Example**
>
> An array of integers called `scores`:
> `int[] scores;`

After you create an array, and before you use it, you have to initialize it. To do this,

the formula is:

```
_arrayName = new type[desired_length] {item1, item2, item3, ...};
```

Bear in mind, this formula only works if the array is already declared in a previous line.

Example

In the previous example, we declared an array called scores. But it is not initialized. To initialize it with a length of 3 and 50, -80, and 200 as the three items:

```
scores = new int[3] {50, -80, 200};
```

You can do all the steps in the same line of code:

```
type[] _arrayName = new type[desired_length] {item1, item2, item3, ...};
```

Example

```
int[] scores = new int[3] {50, -80, 200};
```

Array[Index]

One of the aspects of arrays that confuses new learners is that the elements in an array are indexed starting from 0. Therefore, an array is indexed from 0 to array.Length - 1.

Example

In the `scores` array:

- The **first** item is **scores[0]** = `50`
- The **second** item is **scores[1]** = `-80`
- The **third** item is **scores[2]** = `200`

To access one element of an array, use brackets and then type the index number in that square bracket.

Example

```
string[] sweets = new string {"cookie", "muffin", "cake",
"lollipop", "hard candy"}
```

- To access `cookie`: `sweets[0]`
- To access `hard candy`: `sweets[4]`

Read the documentation: Arrays - C♯ Programming Guide | Microsoft Docs↗

Array is a collection type. Read the documentation: Commonly Used Collection Types | Microsoft Docs↗
