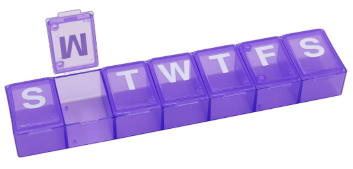
Intro to Arrays Notes

Array is collection of values with the **same datatype**.



The size of the array *(number of elements)* is set when the array defined and cannot be change at run-time.

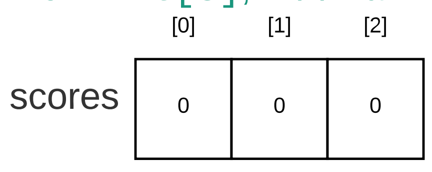
To define an array:

```
datatype[] name-of-array = new datatype[#-of-elements];

int[] nums = new int[10];  // Give me an array with
                           //      10 int elements

double[] charles = new double[3];  // Give me array with
                                   //    3 double elements
```

the datatype of an array includes the **[]**

an element is an item in the array.

```
int[] scores = new int[3];  // array of 3 ints
```

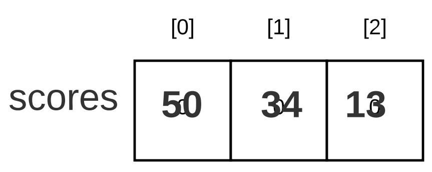


each element in the array is indentified by a number called index starting at 0.

If you don't initialize an array: numerics are set to 0, non-numerics set to null booleans are initialized to false

Use the index inside **[]** to reference an element in the array:

scores[1] = 34;
scores[0] = 50;
scores[2] = 13;



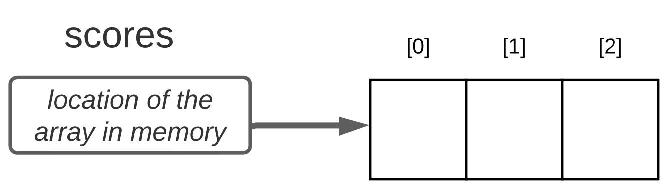
scores[1] - go to scores and move over 1 element
scores[0] - go to scores and move over 0 elements
scores[4] - go to scores and move over 4 elements - error outside array

datatype name = initial-value

```
int[] scores = new int[3];  // array of 3 int
```

`int[] scores` - gives us a variable called scores
`new int[3]` - give us anonymous memory to hold 3 ints and puts the location of the memory in the variable

Array uses two pieces of memory: 1 for the arrayname and one for the elements



array name gets you to the start of the array
[index] tells it now many elements to move over from the start of the array

scores[1] - go to the scores and move over 1 element
scores[0] - go to scores and move over 0 elements

If you use an index that is out side the array - ArrayIndexOutOfRange error.

scores[4] - Array Index exception
scores[-1] - Array Index execption

`array-name.Length` - a property that returns the # elements in the array

The last valid index is always **`array-name.Length - 1`**