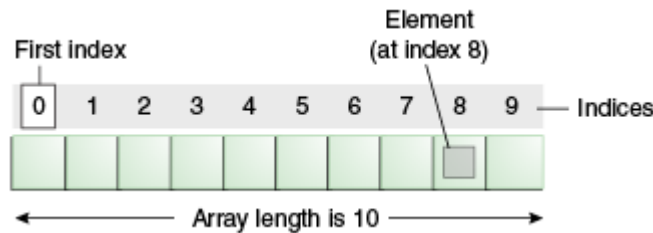


## JAVA Notes 5 – Arrays

---

An array is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed.



Each item in an array is called an element, and each element is accessed by its numerical index. As shown in the picture above, numbering begins with 0. The 9th element, for example, would therefore be accessed at index 8.

### Characteristics of an Array

Here are the characteristics of arrays in Java:

- Fixed length: Once an array is created, we cannot change its size.
- An array can hold primitives (int, char, float) or objects (to be discussed later).
- In Java, the position of an element is specified by index which is zero-based. That means the first element is at index 0, the second element at index 1, and so on.
- An array itself is actually an object.

### Creating arrays:

```
int[] numbers = new int[10];
```

This declares an array object to hold 10 integer numbers (primitive array). When declaring an array of primitive type, one way how we can initialize values for each element of the array is like this:

```
numbers[0] = 10;  
numbers[1] = 500;  
numbers[2] = 1000;  
...
```

We can also declare and initialize elements of an array in one statement. For example:

```
int[] numbers = {10, 500, 1000};
```

This creates an array with 3 integer numbers which are initialized upon declaration. This is a handy shortcut for declaring arrays whose elements are already known before running the program.

The following statement declares an array of String objects:

```
String[] names = new String[5];
```

This statement declares and initializes an array of Strings in one statement. Note that the size of the array is established by the number of items that are declared within the curly brackets.

```
String[] names = {"Paul", "Jane", "Carl", "Mary"};
```

**NOTE:** In Java, you can place the brackets [] either after the type or after the variable name. Hence these declarations are both correct:

```
String[] names = new String[5];  
String names[] = new String[5];
```

However, it's recommended to use the [] after the type for readability.

### Inputting data into an array

It is very common to use the **for** loop to enter data into an array. For example, the following enters 10 exam marks in an array named 'marks'.

```
for (int i = 0; i<10; i++){  
    System.out.print("Enter mark: ");  
    marks[i] = Keyboard.readInt();  
}
```

Again, using the **for** loop we can output the data from the array to the screen. The following code outputs the marks that were entered into the array through the above code:

```
for (int i = 0; i<10; i++){  
    System.out.print(marks[i]);  
}
```