Array: An array is a collection of similar types of data.

Example: If we want to store the names of 11 players of Cricket or 100 names of peoples of a village then we can create an array of the string type that can store 11 names of players and 100 names of people of a village.

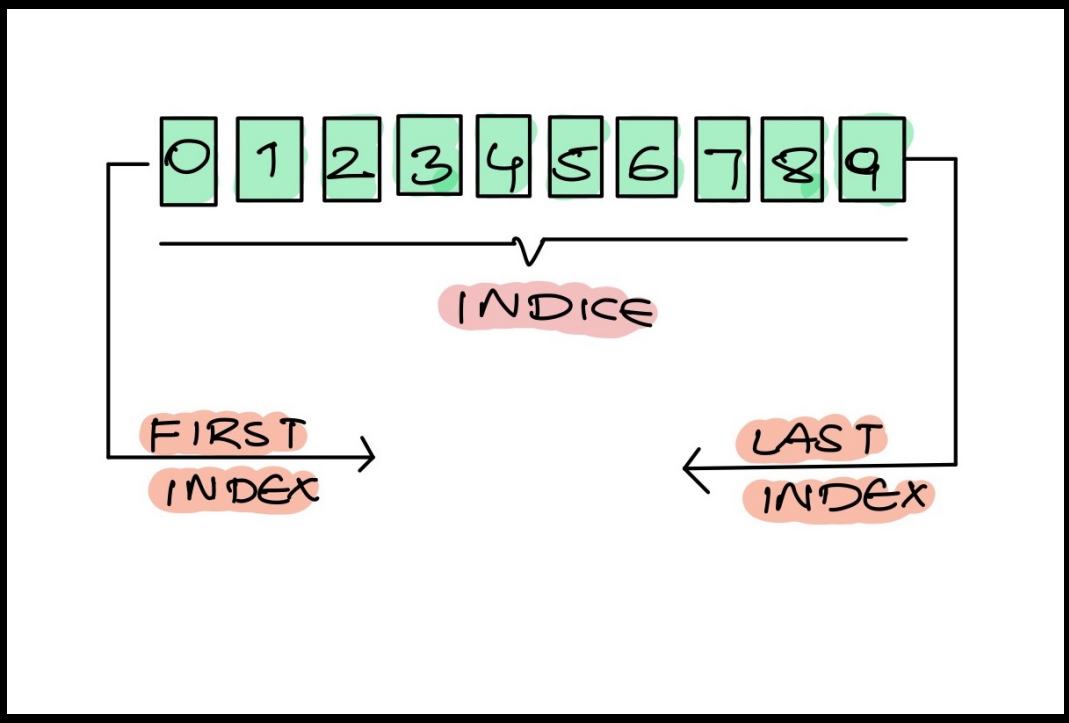
```
String[] array = new String[100];
```

Here, The above array cannot store more than 100 names. The number of values in a Java Array is fixed.

* **Java array** is an object which contains elements of a similar data type. Additionally, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

✧Array in java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and soon.

☆ Unlike C/C++, we can get the length of the array using the length member. In C/C++, we need to use the sizeof operator.



Array Index

Advantage of Array:

1. Array provides the single name for the group of variables of the same type therefore, it is easy to remember the name of all the elements of an array.
2. Traversing an array is a very simple process, we just need to increment the base address of the array in order to visit each element one by one.

3. Any element in the array can be directly accessed by using the index.

Syntax:

```
int[] roll = {1,2,3,4,5,6,7,8,9,10};
```

oR, We can also write also this

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

Where:

int[]: represent the type of data stored in Array.

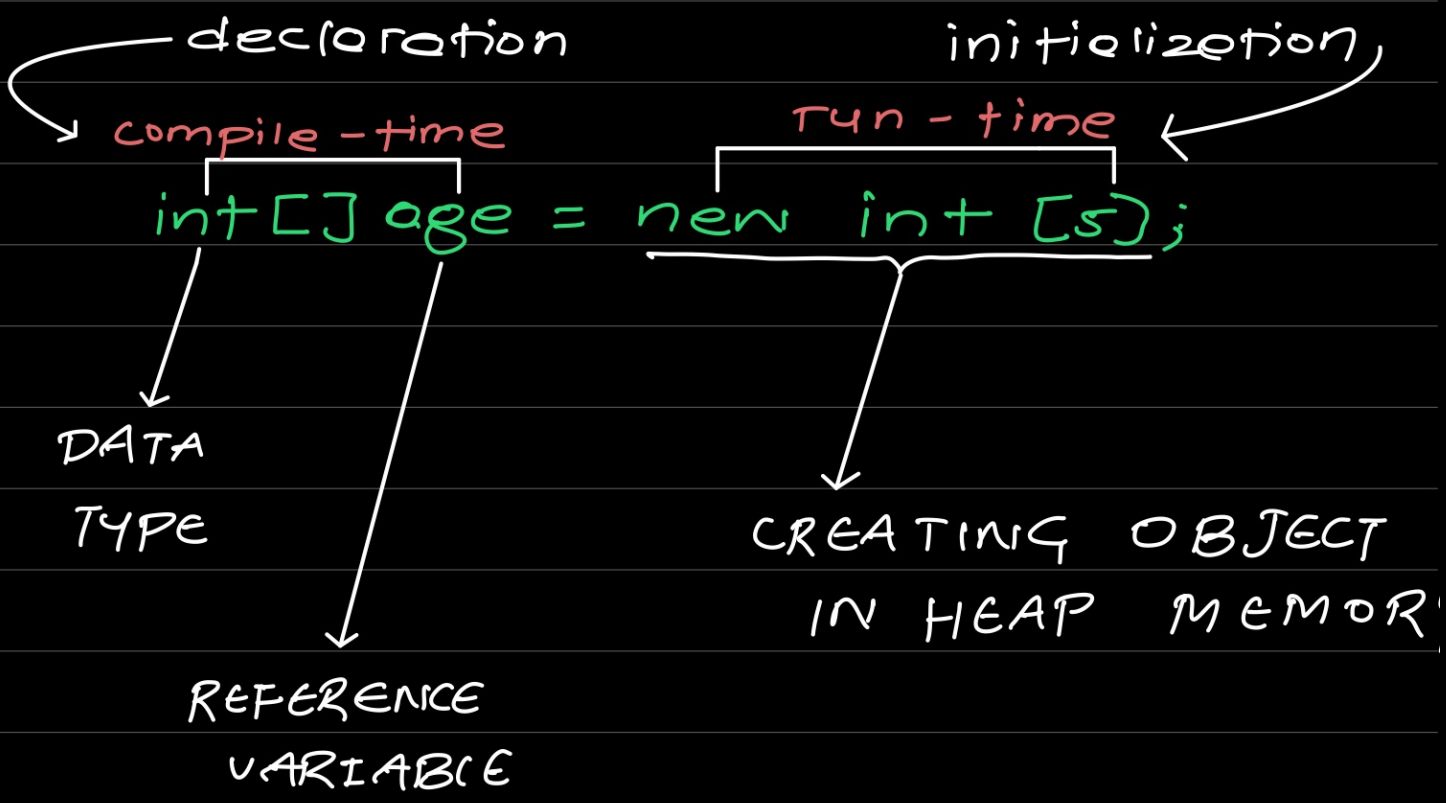
{....}: numbers, all the type of data in array should be same.

Internal working of Array:

int[] roll: declaration of array roll are getting defined in stack.

int[] roll = new int[]: initialization, actual memory allocation happens here, object is being created in heap memory.

ARRAY :-

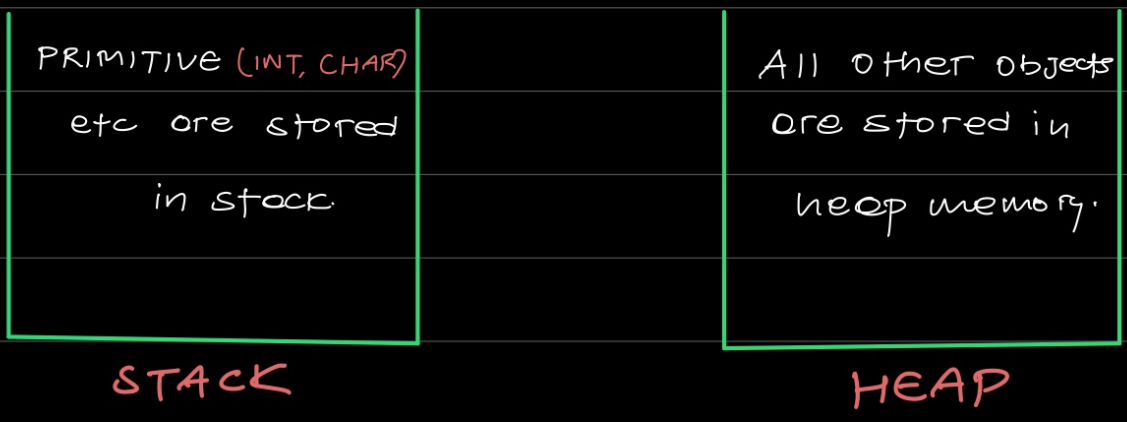


→ This above concept is known as dynamic memory allocation which means at runtime or execution time memory is allocated.

*. Objects are stored in heap memory.

NEW KEYWORD :-

- used to create on object
- it will create on object in heap memory.
- IF we don't provide values in the array internally by default it stores [0,0,0,0,0].



Two Dimensional Array: A 2D array can be visualised as a matrix.

oR

Data is stored in row and column based index (also known as matrix form).

SYNTAX:

```
int[][] arr = new int[3][3]; // 3 rows and 3 columns
```

oR

```
int[][] arr = {
                {1,2,3,4}
                {2,4,6,8}
                {3,6,9,12}
                {4,8,12,16}
                {5,10,15,20}
              }
```

* Upper array are 5(rows) * 4(column).

Dynamic Array:

```
int[][] num = {
                  {1,2,3,4}
                  {2,4,6}
                  {4,8}
                  {5,10,15,20,25} }
```

* Upper array is dynamic array because the number of rows is 4 but the number of column is not fixed (dynamic).

Three Dimensional Array

```
int[][][] intArray = new int[10][20][30];//3D Array
```

Programs:
https://github.com/thedhruvlenka/Java/tree/main/Array