



Array

- Array is a collection of similar type of elements that have a contiguous memory location.
- An array is a very common type of data structure with the elements of same datatype.
- Once defined, the size of an array is fixed and cannot increase to accommodate more elements, index starts from zero to n-1.
- Can store only the fixed size of elements in the array. It doesn't grow its size at runtime.

Array Declaration

`int[] arr=new int[5];` → empty array declared with size

`int[] arr={1,2,3,4,5};` → declared with elements;

`int arr[]={1,2,3,4,5};` → declared with elements

To add elements to the empty array

`int[] arr=new int[5]; String[] words=new String[2];`

`arr[0]=1; words[0]="Learn";`

`arr[1]=2; words[1]="Java";`

`arr[2]=3;`

`arr[3]=4;`

`arr[4]=5;`

To find the length of array

`arr.length;` → returns size of the array as integer

To print an array

Array cannot be printed directly, to print the array elements

`for(int i=0;i<=arr.length;i++)`

`{ System.out.println(arr[i])}`

The Other way to print an array

`System.out.println(Arrays.toString(arr));`

To sort an array

`Arrays.sort(arr);`

***Note :** Array can be sorted in ascending order only.

Two types of array.

- **Single Dimensional Array** – Example: `int [] array1 = {1, 2, 3, 4};`
- **Multidimensional Array** – Example: `int [] [] array2 = {{1, 3, 5}, {2, 4, 6}}`

String

Strings are a non-primitive data type that represents a sequence of characters.

- String type is used to declare string variables.
- Java strings are immutable; we cannot change them.
- Whenever a string variable is created, a new instance is created.

String Declaration

Using Literal	Using new operator
String word="Java";	String word= new String ();

String Methods

Methods	Description
word.length();	to determine the length of the String
word.equals("java");	to compare the contents of two strings
word.equalsIgnoreCase("selenium")	to compare the contents of two string irrespective of the cases
word.concat("selenium")	to append two strings
word.charAt(int index)	returns the character at the given index position
word.toCharArray();	Converts the string into character array.
word.contains("ja")	to search for the match in the given string and it returns boolean
word.toLowerCase();	to convert the whole string in lowercase
word.toUpperCase();	to convert the whole string in uppercase
word.split();	to split the string into array based on delimiter
word.substring(int beginningIndex) word.substring(int beginningIndex, int endIndex)	*returns the string value from given index *returns the string value from given range of index(excludes the character of last index)
word.indexOf(char); word.lastIndexOf(char);	*returns the first index position of the given character *returns the last index position of the given character

***Note :** No reverse method is available in reverse the string