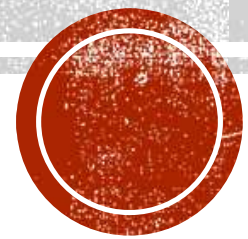
ARRAY IN JAVA



WHAT IS ARRAY

1. Array is a group of primitive data type values or objects.
2. Array is group of similar data type of values or object.
3. Array is a fixed in size.
4. For creating an Array you must know the data type and the size of the values.
5. Array is a non-primitive data type.
6. Array is indexed based.
7. Indexes are always start from 0.
8. Indexes are always managed internally by java.
9. To set and get the values form array you have to use index.
10. There are multiple types array in java such as 1-D array, 2-D, Multi-Dimensional (Jagged Array) Array
11. There are 3 steps to create array
    a. Array Declaration
    b. Array Instantiation (Object creation)
    c. Array Initialization.

1-D ARRAY

1. Data Will be store in the form of row.
2. Using length function you can get the total number of values (size) present inside array.
3. You can also find the last index of the array using length function.

Last index = array.length - 1

Syntax:

Array Declaration

DataType identifier[];

Array Instantiation (Object creation)

Identifier = new DataType[Size];

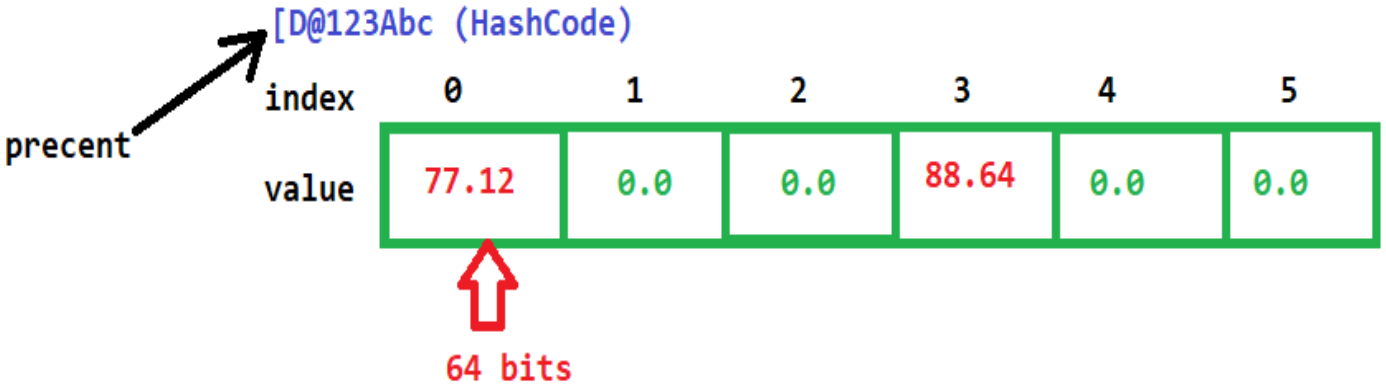
Array Initialization.

Identifier[index] = value;

```
double percent[]; // declaration of Array

percent = new double[6];  // instance creation of array

// Initialization of array
percent[0] = 77.12;
perdent[3] = 88.64;
```



1-D ARRAY

Different ways to create 1-D Array

```
double percent[]; // declaration of array
percent = new double [6]; // instance/Object creation
percent [0] = 77.28;  // Initialization of array
percent[1]= 88.76;
```

```
double percent[] = new double[6];  // declaration and instance/Object creation of array
percent[0] = 77.28;  // Initialization of array
percent[1]= 88.76;
```

```
double percent[] = new double[] {77.28, 0.0, 88.66, 56.65, 77.88, 71.12};
```

```
double percent[] = {77.28, 0.0, 88.66, 56.65, 77.88, 71.12};
```

- **Task**

Create an array which store the marks of 5 subjects. Calculate and print the percentage.
Additional Req: Print the Percent and the grade (Distinction, 1st class ,2nd class ,pass class or fail)

2-D ARRAY

1. Can store a data in the form of row and column.
2. It is also known as matrix.
3. Here, length function returns the total number of rows in array.
   - **array.lengh : return the total number of rows**
   - **array[row_index].length : return the total number of values in a row**

Syntax:

Array Declaration

DataType identifier[][];

Array Instantiation (Object creation)

Identifier = new DataType[ROW_Size][COLUMN_Size];

Array Initialization.

Identifier[row_index][column_index] = value;

```
double percent[][]; // declaration of variable
precent = new double[4][6]; // Instance Creation

precent[0][2] = 77.23;
precent[2][4] = 81.2;
```

Student \ precent	0	1	2	3	4	5
0	0.0	0.0	**77.23**	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	**81.2**	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0

2-D ARRAY

Different ways to create 2-D Array

```
double percent[][]; // declaration of variable
percent = new double[4][6]; // Instance Creation
percent[0][2] = 77.23;  // initialization  of array
percent[2][4] = 81.2;
```

```
double percent[][] = new double[4][6]; // declaration and Instance Creation
percent[0][2] = 77.23;  // initialization  of array
percent[2][4] = 81.2;
```

```
double percent[][] = new double[][] { {56, 67, 56, 67, 87, 67}, {56, 68, 45, 28, 34, 87},
 {84, 75, 87, 34, 81, 34}, {76, 87, 98, 89, 35, 23} };
```

```
double percent[][] = { {56, 67, 56, 67, 87, 67}, {56, 68, 45, 28, 34, 87},
        {84, 75, 87, 34, 81, 34}, {76, 87, 98, 89, 35, 23} };
```

- **Task**

Create an array which store the 4 students 5 subject marks. Calculate and print the percentage.
Also Print the highest percent.
Additional Req: Print the Percent and the grade (Distinction, 1st class ,2nd class ,pass class or fail)

MULTI-DIMENSIONAL ARRAY

1. Can store a data in the form of row and column.
2. In the multi-dimensional array the column size can be dynamically allocated.

Syntax:

Array Declaration

DataType identifier[][];

Array Instantiation (Object creation)

Identifier = new DataType[ROW_Size][];

Array Initialization.

Identifier[row_index][column_index] = value;

Example

```
int array[][] = new int[3][];
array[0] = new int[3];
array[1] = new int[2];
array[2] = new int[5];
array[0][0] = 20;
array[1][1] = 10;
array[0][2] = 25;
array[2][4] = 35;
array[2][1] = 45;
```

Row-0	20	0	25		
Row-1	0	10			
Row-2	0	45	0	0	35