Day-8

**Java Arrays-**

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

To declare an array, define the variable type with square brackets:

String[] cars;

We have now declared a variable that holds an array of strings. To insert values to it, you can place the values in a comma-separated list, inside curly braces:

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

To create an array of integers, you could write:

int[] myNum = {10, 20, 30, 40};

**Access the Elements of an Array**

You can access an array element by referring to the index number.

This statement accesses the value of the first element in cars:

**Example-**

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

System.out.println(cars[0]);

// Outputs Volvo

**Change an Array Element-**

To change the value of a specific element, refer to the index number:

**Example-**

cars[0] = "Opel";

ExampleGet your own Java Server

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

cars[0] = "Opel";

System.out.println(cars[0]);

// Now outputs Opel instead of Volvo

**Array Length-**

To find out how many elements an array has, use the length property:

**Example-**

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

System.out.println(cars.length);

// Outputs 4

**Java Arrays Loop**

You can loop through the array elements with the for loop, and use the length property to specify how many times the loop should run.

The following example outputs all elements in the cars array:

**Example-**

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

for (int i = 0; i < cars.length; i++) {

System.out.println(cars[i]);

}

**Loop Through an Array with For-Each**

There is also a "for-each" loop, which is used exclusively to loop through elements in arrays:

**Syntax-**

for (type variable : arrayname) {

...

}

The following example outputs all elements in the cars array, using a "for-each" loop:

**Example-**

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};

for (String i : cars) {

System.out.println(i);

}

Java Multi-Dimensional Arrays –

**Multidimensional Arrays-**

A multidimensional array is an array of arrays.

Multidimensional arrays are useful when you want to store data as a tabular form, like a table with rows and columns.

To create a two-dimensional array, add each array within its own set of curly braces:

**Example-**

int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };

myNumbers is now an array with two arrays as its elements.

**Access Elements**

To access the elements of the myNumbers array, specify two indexes: one for the array, and one for the element inside that array. This example accesses the third element (2) in the second array (1) of myNumbers:

**Example-**

int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };

System.out.println(myNumbers[1][2]); // Outputs 7

**Change Element Values-**

You can also change the value of an element:

**Example-**

int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };

myNumbers[1][2] = 9;

System.out.println(myNumbers[1][2]); // Outputs 9 instead of 7

**Loop Through a Multi-Dimensional Array**

We can also use a for loop inside another for loop to get the elements of a two-dimensional array (we still have to point to the two indexes):

**Example-**

public class Main {

public static void main(String[] args) {

int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };

for (int i = 0; i < myNumbers.length; ++i) {

for(int j = 0; j < myNumbers[i].length; ++j) {

System.out.println(myNumbers[i][j]);

}

}

}

}