**Array-**

Array is a collection of similar type of elements which has contiguous memory location.

Java array is an object which contains elements of a similar data type.

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, and 2nd element is stored on 1st index and so on.

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



### Advantages

* Code Optimization: It makes the code optimized, we can retrieve or sort the data efficiently.
* An array can store multiple values in a single variable.
* Arrays are fast as compared to primitive data types.
* Members of the array are stored in consecutive memory locations.
* Random access: We can get any data located at an index position.

### Disadvantages

* Size Limit: We can store only the fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in Java which grows automatically.
* They can only store data of a single type.

### There are two types of array as-

* Single Dimensional Array
* Multidimensional Array

### Syntax to Declare an Array in Java

1. dataType[] arr; (or)
2. dataType []arr; (or)
3. dataType arr[];

Instantiation of an Array in Java

1. arrayRefVar=new datatype[size];

### Example of Java Array

Let's see the simple example of java array, where we are going to declare, instantiate, initialize and traverse an array.

//Java Program to illustrate how to declare, instantiate, initialize

//and traverse the Java array.

class Testarray{

public static void main(String args[]){

int a[]=new int[5];//declaration and instantiation

a[0]=10;//initialization

a[1]=20;

a[2]=30;

a[3]=40;

a[4]=50;

//traversing array

for(int i=0;i<a.length;i++)//length is the property of array

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

}}

Output-

10

20

30

40

50

We can declare, instantiate and initialize the java array together by:

int a[]={10,20,30};//declaration, instantiation and initialization

//Java Program to illustrate the use of declaration, instantiation

//and initialization of Java array in a single line

class Testarray1{

public static void main(String args[]){

int a[]={10,20,30};//declaration, instantiation and initialization

//printing array

for(int i=0;i<a.length;i++)//length is the property of array

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

}}

Output-

10

20

30

### For-each Loop for Java Array

We can also print the Java array using [for-each loop](https://www.javatpoint.com/for-each-loop). The Java for-each loop prints the array elements one by one. It holds an array element in a variable, then executes the body of the loop.

The syntax of the for-each loop is given below:

for(data\_type variable:array){

//body of the loop

}

Let us see the example of print the elements of Java array using the for-each loop.

**//Java Program to print the array elements using for-each loop**

class Testarray1{

public static void main(String args[]){

int arr[]={10,20,30,40};

//printing array using for-each loop

for(int i:arr)

System.out.println(i);

}}

Output-

10

20

30

40

**Properties file in Java?**

Properties file is nothing but simple configuration file called as .property file or it is the simple file with .properties extension and it contain the key value formatted data.

The **properties** object contains key and value pair both as a string. The java.util.Properties class is the subclass of Hashtable.

It can be used to get property value based on the property key. The Properties class provides methods to get data from the properties file and store data into the properties file. Moreover, it can be used to get the properties of a system.

**Why?**

In standalone or web application, data is frequently changing is like username, password, url, driver, etc. so every time it is very difficult to modify in java that becomes very complex. So to overcome this problem, we should go for .properties file.

**Recompilation is not required if the information is changed from a properties file**: If any information is changed from the properties file, you don't need to recompile the java class. It is used to store information which is to be changed frequently.

Simple.properties

username=velocity

password=pune

**Program for properties file in Java.**

**package** com.array;

**import** java.io.FileInputStream;

**import** java.util.Properties;

**public** **class** TestArray {

**public** **static** **void** main(String[] args) {

**try** {

FileInputStream fis = **new** FileInputStream("C:\\Users\\ThisPC\\Desktop\\simple.properties");

// locate the properties file

Properties p= **new** Properties();

p.load(fis);

// load properties file by using load () of properties class

String username = p.getProperty("username");

String password = p.getProperty("password");

// get the data by using getProperty()

System.***out***.println("Username is=" + username);

System.***out***.println("Password is=" + password);

} **catch** (Exception e) {

e.printStackTrace();

}

// print data on console

}

}

Output-

Username is=root

Password is=root

Driver is=com.mysql.jdbc.driver