**Java Loop statements**  
Whenever we want to execute a block of statements several times then we use Loop structures.  
  
**There are four types of loop structure in Java.**  
1) for Loop  
  
2) while Loop  
  
3) do...while Loop  
  
4) Enhanced for Loop  
-------------------------  
**1) for Loop**  
It repeats a block of statements for a specified number of times.  
  
**Syntax:**  
for(stratvalue; endValue; increment/decrement) {  
Statements  
---------  
--------  
}  
  
**Example:**  
for (int i=1; i<=10; i++)  
        {  
            System.out.println(i);  
        }  
**Example 2:**  
// print 10 to 1 Numbers using For Loop.  
          
        for (int i=10; i>=1; i--)  
        {  
            System.out.println(i);  
        }  
**Example 3:**// print every 10th Number up to 100.  
          
        for (int i=10; i<=100; i=i+10)  
        {  
            System.out.println(i);  
        }  
--------------------------------------  
**2) while Loop**  
It repeats a block of statements while condition is true.  
  
**Syntax:**  
Initialization  
while (condition) {  
statements  
------  
-------  
increment/decrement  
}  
  
**Example:**// print 1 to 10 Numbers using While Loop.  
          
        int i = 1;  
        while (i <=10){  
            System.out.println(i);  
            i++;  
        }  
**Example 2:**// print 10 to 1 Numbers using While Loop.  
          
        int i = 10;  
        while (i >= 1){  
            System.out.println(i);  
            i--;  
        }  
**Example 3:**// print every 10th number up to 100 using While Loop.  
          
        int i = 10;  
        while (i <= 100){  
            System.out.println(i);  
            i= i+10;  
        }  
--------------------------------------  
**3) do ...while Loop**---------------------------------------  
it repeats a block of statements while condition is true and Irrespective of the condition, it executes a block of statements at least once.  
  
**Syntax:**  
do   
{  
Statements  
----------  
----------  
Increment/Decrement  
} while(Condition);  
------------  
**Example:**int i = 20;  
        do {  
         System.out.println("i value is : " + i);  
                i++;  
        } while(i <= 10);  
---------------------------------------  
**4) Enhanced for Loop**---------------------------------------  
It executes all elements in an Array  
  
**Syntax:**  
for (declaration: expression) {  
statements  
----------  
-----------  
}  
  
**Example:**String languages [] = {"C", "C++", "Java", "COBOL"};  
        for (String lang: languages)  
        {  
            System.out.println(lang);  
        }  
------------------------

**String Handling in Java**  
  
String is sequence of characters written in double quotes.  
  
I) Java String Example:   
package javaExamples;  
  
class StringExample {  
public ststic void main (String [] args) {  
  
System.out.println ("Hello Selenium"); // Hello Selenium is a String  
System.out.println ("123 Hello Selenium");  
System.out.println ("Hello Selenium 123");  
System.out.println ("123\*456 Hello Selenium");  
}  
}  
Note: String may contain Alfa bytes, numbers and special characters.  
Creating Strings  
  
> String is considered as object in java.  
  
II) Example for Creating Strings:  
package javaExamples;  
  
public class StringExample {  
      
public static void main (String []args){  
    String myTool = "Selenium"; // String Variable  
  
    String [] myTools = {"UFT", "Selenium", "LoadRunner", "Quality Center"}; // Array of Strings.  
  
    System.out.println(myTool);// Selenium  
  
    for (int i=0; i < myTools.length; i++) {  
        System.out.println(myTools[i]); // Print Array of strings      
        }  
    }  
}  
--------------------------------------  
III) Concatenating Strings  
package javaExamples;  
  
public class StringExample {  
      
public static void main (String []args){  
    String str1 = "Selenium"; // String Variable  
    String str2 = "Testing";  
  
    System.out.println(str1 + str2);// Selenium Testing  
    System.out.println("Test Automation " + "Using Selenium" + " and Java"); // Test Automation using Selenium and java  
    System.out.println(1 + 1 + " Selenium");  
    System.out.println("Selenium" +1 + 1);  
        }  
}  
-------------------------------  
IV) String Comparison in Java:  
a) String comparison using equals() method  
  
b) String comparison using == (Relational Operator)  
  
c) String comparison using compareTo() method  
  
Example:  
  
package javaExamples;  
  
public class StringExample {  
      
public static void main (String []args){  
    String str1 = "SELENIUM";   
    String str2 = "selenium";  
    String str3 = "SELENIUM";  
    String str4 = "selenium testing";  
      
    System.out.println(str1.equals(str2));// false  
    System.out.println(str1.equals(str3));// true  
      
    System.out.println(str1 == str2); // false  
    System.out.println(str1 == str3); // true  
}  
}  
--------------------------  
**V) Important operations on Strings:**  
   
package javaExamples;  
  
public class StringExample {  
      
public static void main (String []args){  
       String str1 = "Selenium";  
       String str2 ="abcdfg@gmail.com";  
         
       System.out.println(str1.length());// 8 **(finding length of the String)**  
        
       System.out.println(str1.contains("len"));// true **(finding sub string)**  
       System.out.println(str1.contains("lem"));// false (finding sub string)  
        
      **// Returning Sub Strings** System.out.println(str2.substring(0)); //   
       System.out.println(str2.substring(9)); // gmail.com  
       System.out.println(str2.substring(14)); // com  
       System.out.println(str2.substring(9, 13)); // gmail  
    }  
}  
-----------------------------------------

**Arrays in Java**  
   
1) Introduction:  
> Java Array is an object that holds a fixed number of values of a single data type.  
  
> The length of an Array is established when the Array is created.  
  
> Array length is fixed, Java Array has Zero based index.  
  
2) Declaration of Arrays in Java:  
int abc []; // Array of Integers  
  
abc = new int[10]; //creating Array and defning size.  
  
System.out.println(abc.length); //Finding length of the Array.  
  
abc[0] =10;  
abc[1] = 20;  
.  
.  
--------------  
int [] abc = new int[5];  
  
abc[0] =2;  
abc[1] =3;  
System.out.println(abc[0]+abc[1]);  
------------------------------  
int [] xyz = {10, 20, 30, 40, 50} //Creating Array and Initializing   
  
System.out.println(abc.length);  
System.out.println(xyz[1]+xyz[2]);  
//Priting Array  
for (int i=0; i < xyz.length; i++) {  
    System.out.println(xyz[i]);      
    }  
-------------------  
**Creating Arrays (Different data types)**  
int [] array1 ={1, 2, 3, 4, 5}; // Array of Integers  
char [] array2 ={'A', 'B', 'C'}; // Array of Characters  
boolean [] array3 = {true, false, false, true,false}; // Array of Boolean values  
String [] array4 = {"Selenium", "UFT", "Java", "LoadRunner"}; // Array of Strings  
      
    System.out.println(array1[1]); // 2  
    System.out.println(array2[1]); // B  
    System.out.println(array3[1]); // false  
    System.out.println(array4[0]); // Selenium  
      
--------------------------------  
3) Copy of values an Array into another Array  
  
**Example 1:**int [] array1 ={1, 2, 3, 4, 5};  
    int [] array2;  
    array2 = array1;  
      
    System.out.println(array2[2]);  
  
**Example 2:**  
int [] array1 ={1, 2, 3, 4, 5};  
    int [] array2 = new int [array1.length];  
      
    System.arraycopy(array1, 2, array2, 2, 3);  
    System.out.println(array2[2]);  
  
4) Advantages of Arrays:  
   
> Using Arrays we can optimize the code, data can be retrieved or sorted easily.  
  
> We can get required data using index position.  
  
5) Disadvantages of Arrays  
   
> We can store fixed number of elements only in the Array, It doesn't change its size at Runtime.  
  
> An Array holds only one type of data.  
  
6) Types of Arrays  
   
a) Single dimensional Array  
  
b) Multidimensional Array  
  
**Creating multidimensional Array:**  
  
int [][] abc = new int [3][4];  
  
abc [0][0] =1;  
abc [0][0] =2;  
.  
.  
-----------  
int xyz [][] = {{1, 3, 5, 7, 9}, {2, 4, 6, 8, 10}};  
-----------------------------------------------------

**Java Built in Methods**  
  
> Java has a library of classes and methods, organized in packages.  
  
> In order to use built in methods, we import pre-defined packages/Classes.  
  
> java.lang package is automatically imported in any java program.  
  
> Using import keyword we can import pre-defined packages.  
  
Categories of Built in methods:  
I) Number methods  
II) Character methods  
III) String methods  
IV) Array methods etc...  
------------------------------  
**I) Number Methods**  
   
**1) compareTo() Method**  
Example:  
Integer a = 5;  
  
System.out.println(a.compareTo(8)); //-1  
System.out.println(a.compareTo(5));//0  
System.out.println(a.compareTo(2));//1  
  
**Result Criteria:**  
if the integer is equal to the argument then 0  
if the integer is less than the argument then -1  
if the integer is greater than the argument then 1  
-----------------------------------  
**2) equals() Method**  
Integer a = 5;  
  
Integer b = 10;  
Integer c = 5;  
Short d = 5;  
System.out.println(a.equals(b));//false  
System.out.println(a.equals(c)); //true  
System.out.println(a.equals(d)); //false  
--------------------------------------  
**3) abs Method** (Returns Absolute value)  
  
Example:  
  
Integer a = -5;  
double b = -10.234;  
          
System.out.println(Math.abs(a));// 5  
System.out.println(Math.abs(b));// 10.234  
----------------------------------------------  
**4) round Method** (Rounds the value nearest Integer)  
  
Example:  
double a = 10.575;  
double b = 10.498;  
          
System.out.println(Math.round(a));// 11  
System.out.println(Math.round(b));// 10  
-----------------------------------------  
**5) min Method** (Returns minimum value between two numbers)  
  
Example:  
  
int a =10, b =20;  
double c = 1.234, d = 3.567;  
System.out.println(Math.min(a, b)); // 10  
System.out.println(Math.min(c, d)); // 1.234  
System.out.println(Math.min(123, 124)); // 123  
System.out.println(Math.min(10.345, 10.3451)); // 10.345  
System.out.println(Math.min(1, 1)); // 1  
-----------------------------------------  
**6) max Method** (Returns maximum value between two numbers)  
  
Example:  
int a =10, b =20;  
double c = 1.234, d = 3.567;  
System.out.println(Math.max(a, b)); // 20  
System.out.println(Math.max(c, d)); // 3.567  
System.out.println(Math.max(123, 124)); // 124  
System.out.println(Math.max(10.345, 10.3451)); // 10.3451  
System.out.println(Math.max(1, 1)); // 1  
-------------------------------  
**7) random Method** (Generates Random Number)  
  
Example:  
System.out.println(Math.random()); //   
System.out.println(Math.random()); //   
---------------------------------------------  
**II) Character Methods**  
   
**1) isLetter Method** (Checks weather the value is Alfa byte or not?)  
  
Example:  
char a = '1';  
System.out.println(Character.isLetter(a)); //false  
System.out.println(Character.isLetter('A'));//true  
System.out.println(Character.isLetter('a'));//true  
System.out.println(Character.isLetter('\*'));//false  
--------------------------------  
**2) isDigit Method** (It returns weather the value is Number or not?)  
  
char a = '1';  
System.out.println(Character.isDigit(a)); //true  
System.out.println(Character.isDigit('A'));//false  
System.out.println(Character.isDigit('a'));//false  
System.out.println(Character.isDigit('\*'));//false  
System.out.println(Character.isDigit('7')); //true  
-----------------------------------------------  
**3) isUppercase Method** (Checks weather the value is Upper case or not?)  
  
Example:  
System.out.println(Character.isUpperCase('C'));//true  
System.out.println(Character.isDigit('z')); //false  
-------------------------------------  
**4) isLowercase Method** (Checks weather the value is Lower case or not?)  
  
Example:  
System.out.println(Character.isLowerCase('C'));//false  
System.out.println(Character.isLowerCase('z')); //true  
------------------------------  
**5) toUppercase Method** (Converts the value to Upper case)  
  
Example:  
System.out.println(Character.toUpperCase('a'));//A  
System.out.println(Character.toUpperCase('A')); //A  
-----------------------------------  
**6) toLowercase Method** (Converts the value to Lower case)  
  
Example:  
System.out.println(Character.toLowerCase('a'));//a  
System.out.println(Character.toLowerCase('A')); //a  
-------------------------------------------------------  
**III) String Methods**  
   
**1) compareTo()** Method (It compares two strings)  
  
Example:  
String str1 ="SELENIUM";  
String str2 ="selenium";  
String str3 ="seleniuma";  
String str4 ="selenium";  
int result;  
result = str1.compareTo(str2);  
System.out.println(result); //  
          
result = str3.compareTo(str2);  
System.out.println(result); //  
      
result = str2.compareTo(str4);  
System.out.println(result); //  
-------------------------------------  
**2) charAt Method** (character by position)  
  
String str1 ="SELENIUM";  
char result = str1.charAt(0);  
System.out.println(result); //S  
-------------------------  
**3) concat Method** (String concatenation)  
  
String str1 ="Selenium";  
String str2 = " Java";  
str1 = str1.concat(str2);  
System.out.println(str1);   
-----------------------------  
**4) equals Method** (String equals)  
  
Example:  
String str1 ="Selenium";  
String str2 = "UFT";  
String str3 ="Selenium";  
          
System.out.println(str1.equals(str2)); //false  
System.out.println(str1.equals(str3));     //true      
-----------------------------  
**5) equalsIgnorecase Method**    
  
Examples:  
String str1 ="selenium";  
String str2 = "UFT";  
String str3 ="SELENIUM";  
String str4 ="SELENIUM";  
          
System.out.println(str3.equalsIgnoreCase(str4)); //true  
System.out.println(str1.equalsIgnoreCase(str3)); //true      
System.out.println(str1.equalsIgnoreCase(str2)); //false  
-----------------------------------------------  
**6) toUppercase Method** (Converts values To Upper case)  
  
Example:  
String str1 ="selenium";  
String str2 ="SELEnium";  
String str3 ="SELENIUM";  
          
System.out.println(str1.toUpperCase());  
System.out.println(str2.toUpperCase());  
System.out.println(str3.toUpperCase());  
-------------------------------------  
**7) toLowercase Method** (Converts values To Lower case)  
  
String str1 ="selenium";  
        String str2 ="SELEnium";  
        String str3 ="SELENIUM";  
          
        System.out.println(str1.toLowerCase()); //selenium  
        System.out.println(str2.toLowerCase()); //selenium  
        System.out.println(str3.toLowerCase()); //selenium  
---------------------------------------------  
**8) trim Method** (removes spaces from both sides of a String)  
  
Example:  
  
String str1 ="           Selenium              ";  
String str2 ="               SELEnium";  
String str3 ="SELENIUM                ";  
          
System.out.println(str1);  
System.out.println(str1.trim());   
System.out.println(str2.trim());  
System.out.println(str3.trim());  
---------------------------------------------  
**9) substring Method** (Returns sub string)  
  
Example:  
String str1 ="Welcome to Selenium Testing";  
          
System.out.println(str1.substring(10)); // Selenium Testing  
System.out.println(str1.substring(19)); //Testing  
System.out.println(str1.substring(10, 18)); //Selenium  
------------------------------------------------  
**10) endsWith Method** (ends with specified suffix)  
  
Example:  
-----------  
String str1 = "Selenium Testing";  
          
System.out.println(str1.endsWith("Testing"));//true  
System.out.println(str1.endsWith("ing"));//true  
System.out.println(str1.endsWith("Selenium"));//false  
--------------------------------------------------  
**11) length() Method** (Returns length of a String)  
  
String s = "Selenium";  
System.out.println(s.length()); //8  
--------------------------------------------------  
**IV) Array Methods**  
   
**1) length() Method**  
int [] array1 = {10, 20, 30, 40, 50};  
System.out.println(array1.length);//5  
-------------------------------------  
**2) toString() Method**(print Array)  
  
String [] arr1 ={"UFT", "Selenium", "RFT", "SilkTest"};  
String str = Arrays.toString(arr1);  
System.out.println(str);  
--------------------------------------  
**3) contains() Method**(Checks if the Array contains certain value or not?)  
  
String [] arr1 ={"UFT", "Selenium", "RFT", "SilkTest"};  
boolean a = Arrays.asList(arr1).contains("UFT");  
boolean b = Arrays.asList(arr1).contains("uft");  
  
System.out.println(a);// true  
System.out.println(b);// false  
----------------------------------------------------------