Char Literal :
----
* It is also known as Character Literal.
* In char literal we have only data type i.e char data type which accepts 16 bits of memory.
* Char literal we can represent in the following ways :

a) Single character/digit/symbol enclosed in single quotes.
   Example : char ch = 'A';

b) C language supports ASCII format where range is 0 - 255, On the other hand java language supports UNICODE (ASCII + non ASCII) format where range is 0 - 65535. By using UNICODE format we can represent any language character.
   Example : char ch1 = 65535; //Valid
             char ch2 = 65536; //Invalid

c) We can assign character literal to integral literal to get the UNICODE value of that particular character.
   Example :   int x = 'A';

d) All the escape sequences we can represent as a character literal.
   Example : char ch = '\n';

e) We can also represent char literal in **4 digit hexadecimal number in UNICODE format.**

   The format is : **'\uXXXX'**

   Here \u represents, it is UNICODE and X represents digits [It must be in 4 digit only]

   Minimum Range : '\u0000'
   Maximum Range : '\uffff'

   Example :

   Let suppose I want to represent character **A** in <u>4 digit hexadecimal format.</u>

   The UNICODE value of A is 65.

   $(65)_{10} = (?)_{16}$

16	65	1
16	4	4
	0	

   $(65)_{10} = (041)_{16}$

   -> 041 is in hexadecimal number but It voilates the format becaue the format is '\uXXXX'

   Example :
   char x = '\u0041';
   System.out.println(x); //A

//Programs on Character Literal :

```
public class Test1
{
    public static void main(String[] args)
    {
        char ch1 = 'a';
        System.out.println("ch1 value is :"+ch1);

        char ch2 = '5';
        System.out.println("ch2 value is :"+ch2);

        char ch3 = '@';
        System.out.println("ch3 value is :"+ch3);

    }
}
public class Test2
{
    public static void main(String[] args)
    {
        int ch = 'A';
        System.out.println("ch value is :"+ch);

        char ch1 = 'A';
        System.out.println("ch1 value is :"+ch1);
    }
}
//The UNICODE value for ? character is 63
public class Test3
{
    public static void main(String[] args)
    {
        char ch1 = 63;
        System.out.println("ch1 value is :"+ch1);

        char ch2 = 64;
        System.out.println("ch2 value is :"+ch2);

        char ch3 = 65;
        System.out.println("ch3 value is :"+ch3);

        char ch4 = 01;
        System.out.println("ch4 value is :"+ch4);
    }
}
public class Test4
{
    public static void main(String[] args)
    {
        char ch1 = 65000;
      System.out.println("ch1 value is :"+ch1);

        char ch2 = 0Xadd;
        System.out.println("ch2 value is :"+ch2);

        char ch3 = 51678;
        System.out.println("ch3 value is :"+ch3);
    }
}
```

Note : Here we will not get proper output because the equivalent translator is not available so unable to convert.

```
//Addition of two character in the form of Integer
public class Test5
{
public static void main(String txt[])
  {
    int x = 'A';
  int y = 'B';

    System.out.println(x+y); //131
    System.out.println('a' + 'a');  //194
    System.out.println("a"+"a"); //aa

  }
}
//Range of UNICODE Value (65535) OR '\uffff'
class Test6
{
    public static void main(String[] args)
    {
        char ch1 = 65535;
        System.out.println("ch value is :"+ch1);

        char ch2 = 65536; //error
        System.out.println("ch value is :"+ch2);
    }
}
//WAP in java to describe unicode representation of char in hexadecimal format
public class Test7
{
    public static void main(String[] args)
    {
        int ch1 = '\u0000';
        System.out.println(ch1);

        int ch2 = '\uffff';
        System.out.println(ch2);

        char ch3 = '\u0041';
     System.out.println(ch3); //A

        char ch4 = '\u0061';
        System.out.println(ch4); //a
    }
}
class Test8
{
    public static void main(String[] args)
    {
        char c1 = 'A';
        char c2 = 65;
        char c3 = '\u0041';

        System.out.println("c1 = "+c1+", c2 ="+c2+", c3 ="+c3);
    }
}

class Test9
{
    public static void main(String[] args)
    {
        int x = 'A';
        int y = '\u0041';
        System.out.println("x = "+x+" y ="+y);
    }
}

//Every escape sequence is char literal
class Test10
{
    public static void main(String [] args)
    {
        char ch ='\n';
        System.out.println("Hello");
        System.out.println(ch);

    }
}

public class Test11
{
    public static void main(String[] args)
    {
        System.out.println("Min Value :"+(int)Character.MIN_VALUE);
        System.out.println("Max Value :"+(int)Character.MAX_VALUE);

        int min = '\u0000';
        int max = '\uffff';

        System.out.println("Min Value :"+min);
        System.out.println("Max Value :"+max);

    }
}
```

Boolean Literal :
----
* In Boolean literal we have only data type which is boolean, It is used to represent either true OR false state.
* It accepts only 1 bit of memory OR depends upon JVM impelementation.
  Example :
     boolean isEmpty = true;
     booelan isValid = false;

* Unlike C language we **can't assign** 0 and 1 to the boolean data type because in java 0 and 1 both are representing integral literal.

     boolean isValid = 0; //Valid in C++ but Invalid in Java

* String we cannot asiign to boolean type.

     boolean isTrue = "true"; //Invalid

//Programs on Boolean Literal

```
public class Test1
{
  public static void main(String[] args)
  {
    boolean isValid = true;
    boolean isEmpty = false;

    System.out.println(isValid);
    System.out.println(isEmpty);
  }
}

public class Test2
{
  public static void main(String[] args)
  {
      boolean c = 0; //error
    boolean d = 1; //error
    System.out.println(c);
    System.out.println(d);
  }
}
public class Test3
{
    public static void main(String[] args)
    {
        boolean x = "true";  //error
        boolean y = "false";  //error
        System.out.println(x);
      System.out.println(y);
    }
}
```

String Literal :
----
* String is a predefined class available in java.lang package.
* String a collection of alpha-neumric characters, which can be different combination of character, digits, special symbol and so on. It must be enclosed with double quotes.
* Internally it is character array.
* In java, We can create String object by using following 3 ways :

1) By using String Literal :

   String str = "India";

2) By using new Keyword :

   String str2 = new String("Hyderabad");

3) By using Character Array [Old technique]
----
   char ch[] = {'R', 'A', 'J'};

//Programs on String Literal :

```
//Three Ways to create the String Object
public class StringTest1
{
    public static void main(String[] args)
    {
        String s1 = "Hello World";      //Literal
        System.out.println(s1);

        String s2 = new String("Ravi"); //Using new Keyword
        System.out.println(s2);

        char s3[] = {'H','E','L','L','O'}; //Character Array
        System.out.println(s3);

    }
}
//String is collection of alpha-numeric character
public class StringTest2
{
    public static void main(String[] args)
    {
        String x="B-61 Hyderabad";
        System.out.println(x);

        String y = "123";
        System.out.println(y);

        String z = "67.90";
        System.out.println(z);

        String p = "A";
        System.out.println(p);
    }
}

//IQ
public class StringTest3
{
    public static void main(String []args)
    {
        String s = 15+29+"Ravi"+40+40;
        System.out.println(s);

    }
}
```