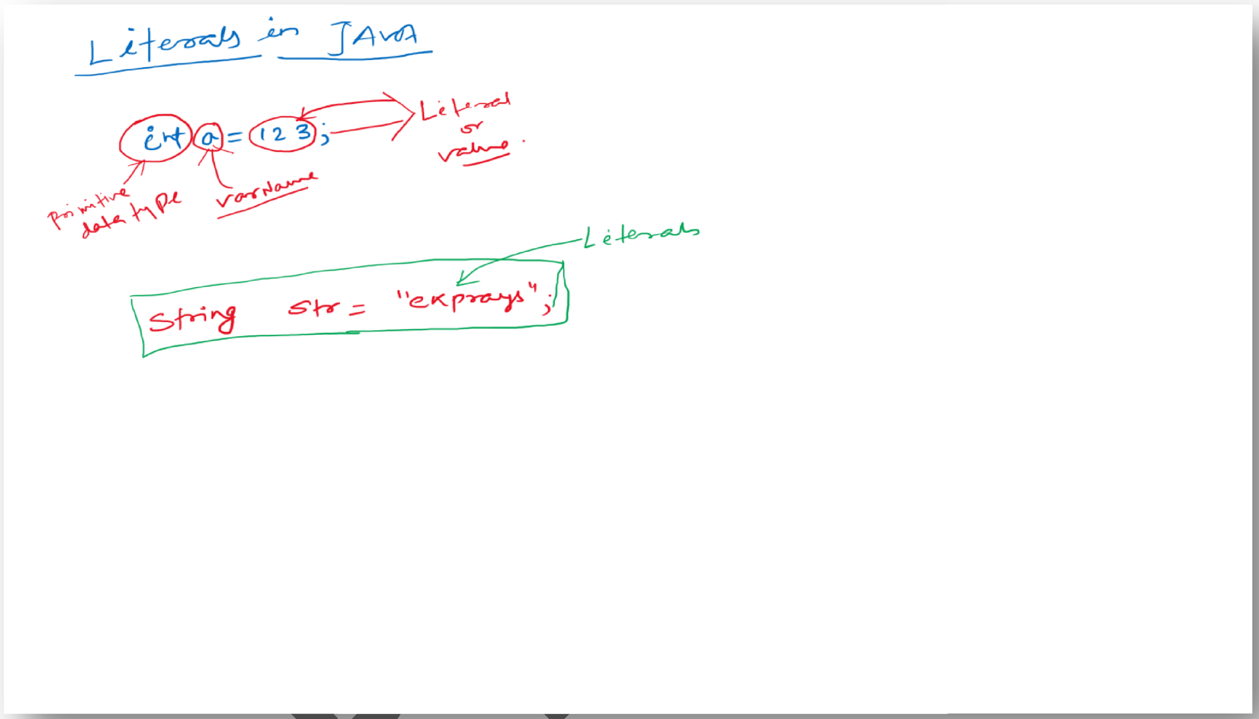
JAVA BASICS

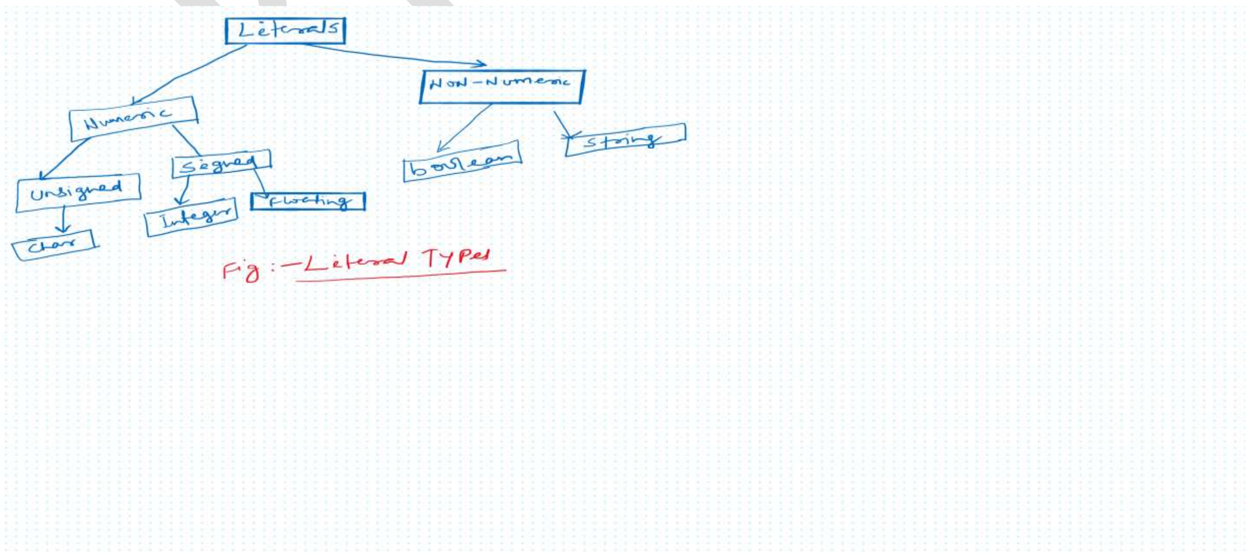
Java -3

11/9/2018
Ranjan Gupta
9971758212

Literals :

- Literals are the actual value.
- Literals will be assigned to variables or constants.
- Literals are also used to perform any operations.
- Constants value in program are called as literals.





Types of Literals

1) Boolean Literals
2) Character Literals
3) String Literals
4) Integral Literals
5) Floating Literals
6) Null Literals

Boolean Literals

Boolean Literals can be assigned to Boolean type of variable.There are two type of Boolean literals

- True
- False.
- `boolean b = true;`

```
Lab31.java
------------------
// Java program to illustrate the application of
boolean
// literals
public class lab31 {
  public static void main(String[] args)
  {
    boolean b = true;
    boolean c = false;
    boolean d = 0;
    boolean b = 1;
    System.out.println(b);
    System.out.println(c);
    System.out.println(d);
    System.out.println(e);
  }
}
```

```
Lab32.java
-----------------
class Lab32{
public static void main(String args[]){
boolean b1=true;
boolean b2=false;
System.out.println(b1);
System.out.println(b2);
}
}
```

```
Lab33.java
-----------------
class Lab33{
public static void main(String args[]){
boolean b = 1;
System.out.println(b);
}
}
```

```
Lab34.java
------------------
class Lab34{
public static void main(String args[]){
boolean b = True;
System.out.println(b);
}
}
```

```
Lab35.java
----------------
class Lab35{
public static void main(String args[]){
boolean True=false;
boolean b= True;
System.out.println(b);
}
}
```

Character Literal:

Character Literal is a single character enclosed within a single quotation mark. Character Literal can be assigned to char type variable.

```
Lab36.java
--------------
class Lab36{
public static void main(String args[]){
char ch1='a';
char ch2='b';
char ch3='5';
char ch4='*';
System.out.println(ch1);
System.out.println(ch2);
System.out.println(ch3);
System.out.println(ch4);
}
}
```

```
Lab37.java
-----------------
class Lab37{
public static void main(String args[]){
char ch=''; // Empty
System.out.println(ch);
}
}
```

```
Lab38.java
-----------------
class Lab38{
public static void main(String args[]){
char ch=' '; // one space
System.out.println(ch);
}
}
```

```
Lab39.java
--------------------
class Lab38{
public static void main(String args[]){
char ch='  '; // Two space
System.out.println(ch);
}
}
```

```
Lab40.java
---------------------
class Lab40{
public static void main(String args[]){
char ch='            '; // tab key
System.out.println(ch);
}
}
```

```
Lab41.java
-----------------------
class Lab41{
public static void main(String args[]){
char ch='AB';
System.out.println(ch);
}
}
```

Lab42.java

```
class Lab42{
public static void main(String args[]){
char ch='\';
System.out.println(ch);
}
}
```

Lab43.java

```
class Lab43{
public static void main(String args[]){
char ch1='+';
char ch2=''';
System.out.println(ch1);
System.out.println(ch2);
}
}
```

Escape Sequence-

Escape sequence is a special notation which is used to represent some special characters which can't be represented as it is.

Escape Sequence	**Description**
\t	Tab space
\b	Backspace
\n	NewLine
\r	Carriage return
\f	Formfeed
\’	Single Quote character
\’’	Double Quote Character
\\\\	Backslash Character.

Lab44.java

```
-----------
class Lab44{
public static void main(String args[]){
char ch='\'';
System.out.println(ch);
}
}
```

Lab45.java

```
-----------
class Lab45{
public static void main(String args[]){
char ch='\\';
System.out.println(ch);
}
}
```

Lab46.java

```
-----------
class Lab46{
public static void main(String args[]){
char ch='\+';
System.out.println(ch);
}
}
```

Lab47.java

```
-----------
class Lab47{
public static void main(String args[]){
char ch='\p';
System.out.println(ch);
}
}
```

```
Lab48.java
-----------
class Lab48{
public static void main(String args[]){
char ch='\a';
System.out.println(ch);
}
}
```

```
Lab48.java
-----------
class Lab48{
public static void main(String args[]){
char ch='"'; // Double Quote in single Quotes
System.out.println(ch);
}
}
```

```
Lab49.java
-----------
class Lab49{
public static void main(String args[]){
char ch='\"'; // slash followed by double quotes
System.out.println(ch);
}
}
```

```
Lab50.java
-----------
class Lab50{
public static void main(String args[]){
String str="welcome to "ekprayas"";
System.out.println(str);
}
}
```

```
Lab51.java
------------------
class Lab51{
public static void main(String args[]){
String str="welcome to \"ekprayas\"";
System.out.println(str);
}
}
```

```
Lab52.java
----------------------
class Lab52{
public static void main(String args[]){
char ch='\u';
System.out.println(ch);
}
}
```

<u>**ASCII character Set:**</u>

- **ASCII stands for American standard code for Information Exchange.**
- Every Character Enclosed in Single quotation marks will have an integer equivalent value called as ASCII value.
- ASCII value range is from 0-255.
- ASCII value can be assigned to a char type variable.

```
Lab53.java
----------------
class Lab53{
public static void main(String args[]){
char ch='1';
System.out.println(ch);
}
}
```

```
Lab54.java
--------------
class Lab54{
public static void main(String args[]){
char ch=1;
System.out.println(ch);
}
}
```

```
Lab55.java
--------------
class Lab55{
public static void main(String args[]){
char ch='1';
System.out.println(ch);
System.out.println(10+ch);
}
}
```

```
Lab56.java
---------------
class Lab56{
public static void main(String args[]){
System.out.println(1+2);
System.out.println('1'+'2');
}
}
```

```
Lab57.java
--------------
class Lab57{
public static void main(String args[]){
char ch1=65;
char ch1=39;
char ch1=23;
System.out.println(ch1);
System.out.println(ch2);
System.out.println(ch3);
}
}
```

```
Lab58.java
--------------
class Lab58{
public static void main(String args[]){
char ch1=260;
char ch2='?';
System.out.println(ch1);
System.out.println(ch2);
}
}
```

```
Lab59.java
--------------
class Lab59{
public static void main(String args[]){
char ch1=0;
char ch2='65535';
System.out.println(ch1);
System.out.println(ch2);
}
}
```

```
Lab60.java
--------------
class Lab60{
public static void main(String args[]){
char ch1=-1;
char ch2='65536';
System.out.println(ch1);
System.out.println(ch2);
}
}
```

UNICODE Characters – Hexadecimal equivalent of decimal

Unicode is a universal international standard character encoding that is capable of representing most of the world's wr
languages.

Why java uses Unicode System?

Before Unicode, there were many language standards:

- **ASCII** (American Standard Code for Information Interchange) for the United States.
- **ISO 8859-1** for Western European Language.
- **KOI-8** for Russian.
- **GB18030 and BIG-5** for chinese, and so on.

Problem

This caused two problems:

1. A particular code value corresponds to different letters in the various language standards.
2. The encodings for languages with large character sets have variable length.
3. Some common characters are encoded as single bytes, other require two or more byte.

Solution

To solve these problems, a new language standard was developed i.e. Unicode System.
In unicode, character holds 2 byte, so java also uses 2 byte for characters.
lowest value:\u0000
highest value:\uFFFF

- UNICODE stands for Universal Code.
- Every character will have Unicode value.

UNICODE Notation

Syntax: \uxxxx -> X is hexadecimal digits
Starts with \u followed by 4 hexadecimal degit.

UNICODE RANGE
\u0000(0) to \uFFFF(65535).

Program:

class Lab62{	**class Lab63{**
public static void main(String args[]){	**public static void main(String args[]){**
char ch1='\u0061';	**int a='\u0061';**
char ch2='\u0062';	**int b='\u0062';**
System.out.println(ch1);	**System.out.println(a);**
System.out.println(ch2);	**System.out.println(b);**
char ch3='\u0041';	**int c='\u0041';**
char ch4='\u0042';	**int d='\u0042';**
System.out.println(ch3);	**System.out.println(c);**
System.out.println(ch4);	**System.out.println(d);**
}	**}**
}	**}**

```
class Lab{
public static void main(String args[]){
int a='\u0037\u0039';
System.out.println(a);
}
}
```

```
class Lab{
public static void main(String args[]){
int \u0061=99;
System.out.println(a);
}
}
```

```
class Lab{
public static void main(String args[]){
int x=\u0061;
System.out.println(x);
}
}
```

```
class Lab{
public static void main(String args[]){
int a=99;
int x=\u0061;
System.out.println(x);
}
}
```

<u>Octal value as Char Type:</u>

Octal value can be assigned to char type.

Octal Notation:

Syntax: -

\DDD – D will be octal digit.

Starts with D followed by three octal digits.

Octal Range-

Range in decimal-> 0 to 255

Range in octal-> \0 to \377.

Note- http://www.asciitable.com/

Program

```
class Lab{
public static void main(String args[]){
char ch1='\101';
char ch2='\103';
System.out.println(ch1);
System.out.println(ch2);
}
}
```

```
class Lab{
public static void main(String args[]){
char ch1=\101;
System.out.println(ch1);
}
}
```

String Literals

Any sequence of characters within double quotes is treated as String literals.

- String Literal is a collection of zero or more characters enclosed between double quotation marks
- String literal can be assigned to reference variable of type string.
- String literal will be represented as char array in memory.

```
String s = "Hello";
// Java program to illustrate the application of String literals
public class Test {
        public static void main(String[] args)
        {
                String s = "Hello";

                // If we assign without "" then it treats as a variable
                // and causes compiler error
                String s1 = Hello;

                System.out.println(s);

                System.out.println(s1);
        }
}
```

Lab76.java
----------------- 	
class Lab 76{
public static void main(String args[]){
String str1=null;
System.out.println(str1);
int x=str1.length();
System.out.println(x);
}
}
	Lab77.java
	--------------------- 	
	class Lab77{
	public static void main(String args[]){
	String str1="";
	System.out.println(str1);
	int x= str1.length();
	System.out.println(x);
	}
	}

```
lab78.java
------------------
class Lab78{
public static void main(String args[]){
String str1="ekprayas@india";
System.out.println(str1);
int x=str1.length();
System.out.println(x);
}
}
```

```
Lab79.java
-----------------
class Lab79{
public static void mian(String args[]){
String str1="ASCII of A is 65";
System.out.println(str1);
String str2="UNICODE of A is \u0041";
System.out.println(str2);
}
}
```

```
Lab80.java
-----------------
class Lab80{
public static void main(String args[]){
String dir1="D:\new\test\batch";
System.out.println(dir1);
String dir2="D:\\new\\test\\batch";
System.out.println(dir2);
}
}
```

```
Lab81.java
--------------------
class Lab81{
public static void main(String args[]){
String dir1="D:\java\core";
System.out.println(dir1);
}
}
```

```
Lab82.java
-------------
class Lab82{
public static void main(String args[]){
String dir1="D:\\java\\core";
System.out.println(dir1);
}
}
```

```
Lab83.java
----------------
class Lab83{
public static void mian(String args[]){
String str1="Octal of 65 is \101";
System.out.println(str1);
String str2="Octal of 65 is \\101";
System.out.println(str2);
}
}
```

SUMMARY

1. **A Boolean type variable can hold either true or false.**
2. **A Boolean type variable can't hold numeric value.**
   a. **0 not equal to false**
   b. **Non zero values not equal to true.**
3. **Boolean literals – true and false are reserved words and must be in lower case.**
4. **A char type variable can hold following.**
   a. **Single character enclosed in single quotation marks.**
   b. **Escape sequence**
   c. **ASCII value**
   d. **UNICODE character**
   e. **Octal Character.**
5. **SPACE is a valid character so it can be placed in single quotation marks.**

6. **Tab is a valid character so it can be placed in single quotation marks.**
7. **Backslash character (\) is used to form escape sequence so only backslash character can’t be placed in single Quotation marks.**
8. **Single quote is used to form character literal so only single quote can’t be placed in single Quotation marks.**
9. **There are some characters like backslash , single quote which can’t be placed in single quotes as it is. Those special characters have to be represented as Escape sequence.**
10. **Every character enclosed in single quotation marks will have an integer equivalent value called as ASCII Value.**
11. **Char type variable can hold integer value ranging from 0 -65535 only.**
12. **Char type variable can hold UNICODE value ranging from \u0000 - \uFFFF only.**
13. **Char type variable can hold OCTAL value ranging from \0 - \377 only.**
14. **When you print the char type variable which holds integer , UNICODE or octal value then it displays corresponding character representation.**
15. **If Integer , UNICODE value don’t have character representation then it displays ? which indicates no character defined for that value.**
16. **When you assign character literals to int type variable then internally ASCII value of that character will be assigned.**
17. **Source code can also be written in UNICODE Representation.**
    a. **int a=\u0037 ; int a=7; -> Valid**
    b. **int a=’\u0037’; int a =’7’; -> Valid**
    c. **int a=\u0037\u0039; int a=79; -> valid**
    d. **int a=’\u0037\u0039’; int a=’79’; -> valid**
18. **Empty String literals is allowed whereas empty character literals is not allowed.**
19. **If reference variable of string type contains null value then it is called as NULL String.**
20. **You can’t refer any members with null string . if you do so it results in NullPointerException.**
21. **If reference variable of string type contains zero character in double quotation marks then it is called as Empty String.**
22. **When java compiler encounters any UNICODE/octal value in String Literals then that UNICODE /OCTAL value will be replaced with actual character.**
23. **You can use escape sequence to represents UNICODE / Octal value as it is in String Literals.**