**Strings:**

A string is a series of characters (It’s an array of characters).

Strings can be mainly created in three ways.

1. Using Character Array

Ex:

char ch = {‘H’,’e’,’l’,’l’,’o’};

String str = new String(ch);

1. Using String Literal

String str1 = “Hello”;

1. Using new keywork

String str2 = new String(“Hello”);

Example:

**package** package1;

**public** **class** JavaExamples {

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

//First way

**char**[] ch = {'H','e','l','l','o','1'};

String str1 = **new** String(ch);

//Second way

String str2 = "Hello2";

//Third way

String str3 = **new** String("Hello3");

System.***out***.println(str1);

System.***out***.println(str2);

System.***out***.println(str3);

}

}

**What is the difference when we create a string with string literal and strings with new keyword?**

When we create a string with string literal, it first searches the memory if there is already an existing string with the same value. If there is an existing string, it just points to that memory.

When we create a string with new keyword, that means we are creating a string object. Even if the string values are same, it always creates a new memory space for the strings and store the string values there.

**Strings are immutable**

Strings are immutable that means they can’t be changed. But they can be assigned to new strings.

This means that when you create a string it will be there in the memory until the object is garbage collected. You can’t change the value of the string in the memory. But when you try to assign it to a new value, that new value will be created in a new memory space. But the old string remains there without any reference.