Java String

String is basically an object that represents a sequence of char values.

An array of characters works the same as Java string.

For example:

1. **char**[] dia={'c','i','t','y'};
    a. String s=**new** String(ch);

       Is same as
2. String s="city";

What is String in Java?

String is a sequence of characters. But in Java, string is an object that represents a sequence of characters.

How to create a string object?

There are two ways to create String object:

1. By string literal
2. By new keyword

1) String Literal

Java String literal is created by using double quotes.

1. String s="welcome";

1. String s1="Welcome";
2. String s2="Welcome";//It doesn't create a new instance

> Note: String objects are stored in a special memory area known as the "string constant pool".

By new keyword

1. String s=**new** String("Welcome");//creates two objects and one reference variable

Java String class provides a lot of methods to perform operations on strings such as compare(),
concat(),
equals(),
split(),
length(),
replace(),
compareTo(),
substring() etc.

Java String Class Methods

Java String toUpperCase() and toLowerCase() method

The Java String toUpperCase() method converts this String into uppercase letter and String toLowerCase() method into lowercase letter.

```
String s="Sachin";
```

```
System.out.println(s.toUpperCase());//SACHIN
System.out.println(s.toLowerCase());//sachin
System.out.println(s);//Sachin(no change in original)
```

Java String trim() method

The String class trim() method eliminates white spaces before and after the String.

```
String s="  Sachin  ";
System.out.println(s);//  Sachin
System.out.println(s.trim());//Sachin
```

Java String startsWith() and endsWith() method

The method startsWith() checks whether the String starts with the letters passed as arguments and endsWith() method checks whether the String ends with the letters passed as arguments.

```
String s="Sachin";
 System.out.println(s.startsWith("Sa"));//true
 System.out.println(s.endsWith("n"));//true
```

Java String charAt() Method

The String class charAt() method returns a character at specified index.

```
String s="Sachin";
System.out.println(s.charAt(0));//S
System.out.println(s.charAt(3));//h
```

Java String length() Method

The String class length() method returns length of the specified String.

```
String s="Sachin";
System.out.println(s.length());//6
```

Java String valueOf() Method

The String class valueOf() method coverts given type such as int, long, float, double, boolean, char and char array into String.

```
int a=10;
String s=String.valueOf(a);
System.out.println(s+10);
```

Java String replace() Method

The String class replace() method replaces all occurrence of first sequence of character with second sequence of character.

```
String s1="Java is a programming language. Java is a platform. Java is an Island.";
```

```
String replaceString=s1.replace("Java","Kava");//replaces all occurrences of "Java" to "Kava"
System.out.println(replaceString);
```

There are three ways to compare String in Java:

1. By Using equals() Method
2. By Using == Operator
3. By compareTo() Method

```
 String s1="Sachin";
String s2="Sachin";
String s3=new String("Sachin");
String s4="Saurav";
System.out.println(s1.equals(s2));//true
System.out.println(s1.equals(s3));//true
System.out.println(s1.equals(s4));//false
```

```
String s1="Sachin";
  String s2="Sachin";
  String s3=new String("Sachin");
```

```
  System.out.println(s1==s2);//true (because both refer to same instance)
  System.out.println(s1==s3);//false(because s3 refers to instance created in nonpool)
```

```
String s1="Sachin";
  String s2="Sachin";
  String s3="Ratan";
  System.out.println(s1.compareTo(s2));//0
  System.out.println(s1.compareTo(s3));//1(because s1>s3)
  System.out.println(s3.compareTo(s1));//-1(because s3 < s1 )
```

String Concatenation in Java

```
String s1="Sachin ";
  String s2="Tendulkar";
  String s3=s1.concat(s2);
  System.out.println(s3);//Sachin Tendulkar
```

Substring in Java

```
String s="SachinTendulkar";
System.out.println("Original String: " + s);
```

```
System.out.println("Substring starting from index 6: " +s.substring(6));//Tendulkar
System.out.println("Substring starting from index 0 to 6: "+s.substring(0,6)); //Sachin
```

https://www.geeksforgeeks.org/strings-in-java/ - why string is immutable