Strings

A sequence of characters is known as strings in java. Double quotes is used to represent a string. Strings are immutable which means once created it cannot be changed.

For Example There are two ways to create a string

1. **String Literal**

   String s = "Java Banters"

2. **Using new keyword**

   String s = new String("Java Banters")

Let's see the difference between the above mentioned ways of creating strings. The JVM maintains a string pool to store all of its strings inside the memory. The string pool helps in reusing the strings.

In example 1 We are just providing the string value directly called "Java Banters" , the compiler checks the string pool first to see if the string already exists.

- **If the string already exists**, the new string is not created. Instead, the new reference, s points to the already existed string.
- **If the string doesn't exist**, the new string (Java Banters) is created.

In example 2 The value of the string is not directly provided. Hence, a new "Java Banters" string is created even though "Java Banters" is already present inside the string pool.

String is a final class in Java . Hence it cannot be extended and this is how it achieves immutability. Due to this String is thread safe and it is secure as some sensitive data cannot be cannot be changed deliberately or accidently.

Comparison of Java Strings

There are three ways to compare strings in java . All three different ways serve it's own purpose

1. By Using equals() method

The String class equals() method compares the original content of the string. It compares values of string for equality. Used for authentication purposes.

2. By Using == Operator

The == operator compares references not values. Used for reference matching

3. By using compareTo() method

The String class compareTo() method compares values lexicographically(alphabetical order) and returns an integer value that describes if first string is less than, equal to or greater than second string.
Suppose s1 and s2 are two String objects. If:

- s1 == s2 : The method returns 0.
- s1 > s2 : The method returns a positive value.
- s1 < s2 : The method returns a negative value.

String Concatenation

String concatenation forms a new String that is the combination of multiple strings. There are two ways to achieve this

1. + (String concatenation) operator
2. concat() method

SubString

A part of String is called substring. There are two ways to get substring from a given String

1. public String substring(int startIndex):
   returns a new string object from the given index (inclusive)

2. public String substring(int startIndex, int endIndex):
   returns a new String object from the specified start index to the end index(exclusive)

Note : IndexOutOfBoundException will be thrown if the index doesn't exists.

Since String is immutable in nature, java provides library classes such as StringBuilder and StringBuffer to work with mutable text data efficiently.

StringBuffer

Java StringBuffer class is thread-safe i.e. multiple threads cannot access it simultaneously. So it is safe and will result in an order.

StringBuilder

The Java StringBuilder class is same as StringBuffer class except that it is non-synchronized.It is not thread safe

StringTokenizer

This class allows you to break a String into tokens. It is simple way to break a String.

For example : The single string "Java Banters" can be broker down into two different string suing the string tokenizer as "Java" and "Banters"