

# OBJECT ORIENTED ANALYSIS & DESIGN DATA STRUCTURES & ALGORITHMS

## String Handling



# Java Strings



# Java Strings

## Object

String is a Java object

## Characters

Represents a sequence of characters

## Class

`java.lang.String` class is used to create and manipulate strings

## Immutable

A string is immutable in nature



# Java Strings

With Strings in Java you can perform various operations, some of which are:

Search

The quick brown fox jumps over the lazy dog

Create Substring

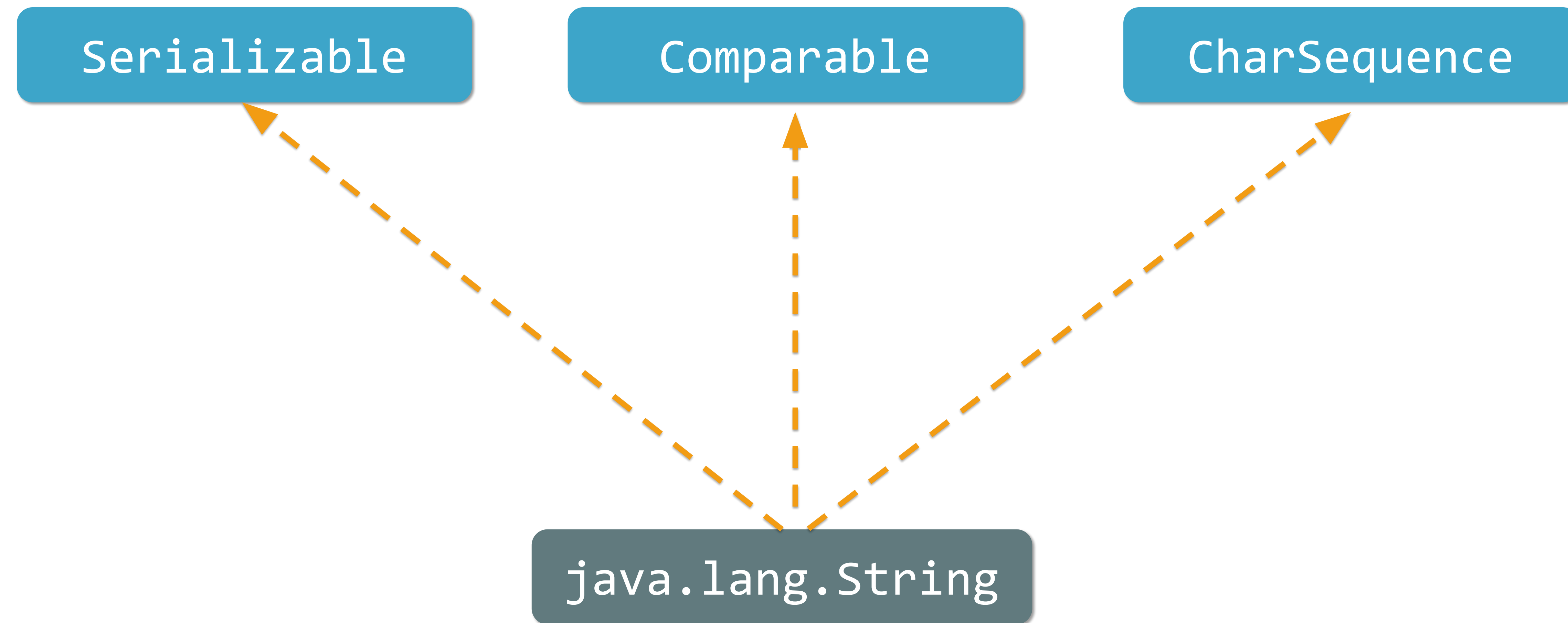
The quick brown fox jumps over the lazy dog

Create new strings

The quick brown fox jumps over the lazy dog



# Java Strings



```
public final class String
    extends Object
    implements Serializable, Comparable<String>, CharSequence
```



# Java Strings

## Serializable

Serializable is a marker interface that contains no data member or method. It is used to “mark” the java classes so that objects of these classes may get a specific capability

## Comparable

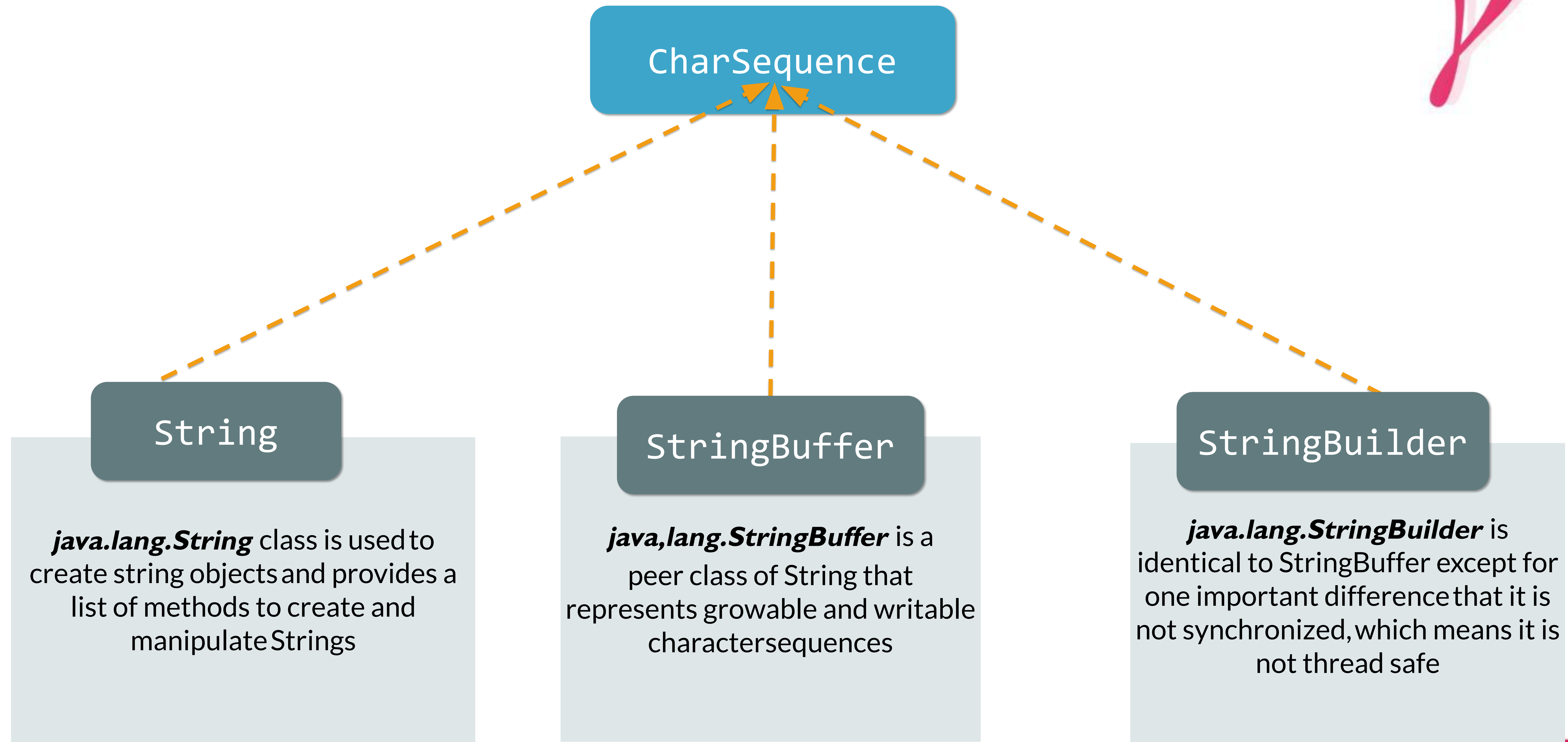
Comparable interface is used for ordering the objects of any user-defined class. This interface is found in **java.lang.package** and contains only one method named **compareTo(Object)**

## CharSequence

A CharSequence interface is a readable sequence of characters. This interface provides uniform, read-only access to various kind of character sequences



# Java Strings – Immutable and Mutable





# Creating a String

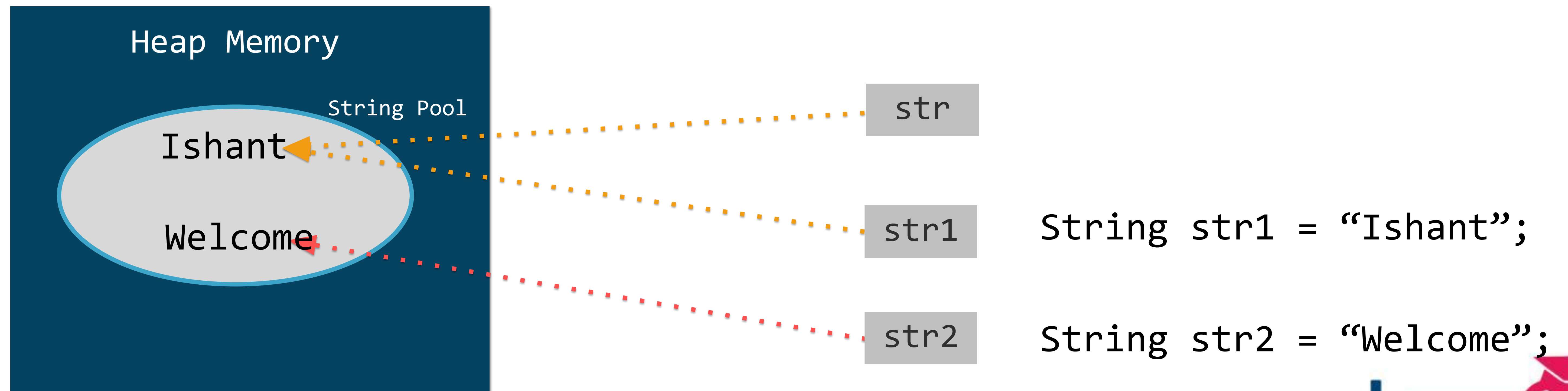


# Creating a String – Using Literal

Java String literal is created by using double quotes

```
String str = "Ishant";
```

Before creating a String literal, Java first looks for a String with same value in the String pool. If found, it returns the reference; else it creates a new String in the pool & returns the reference

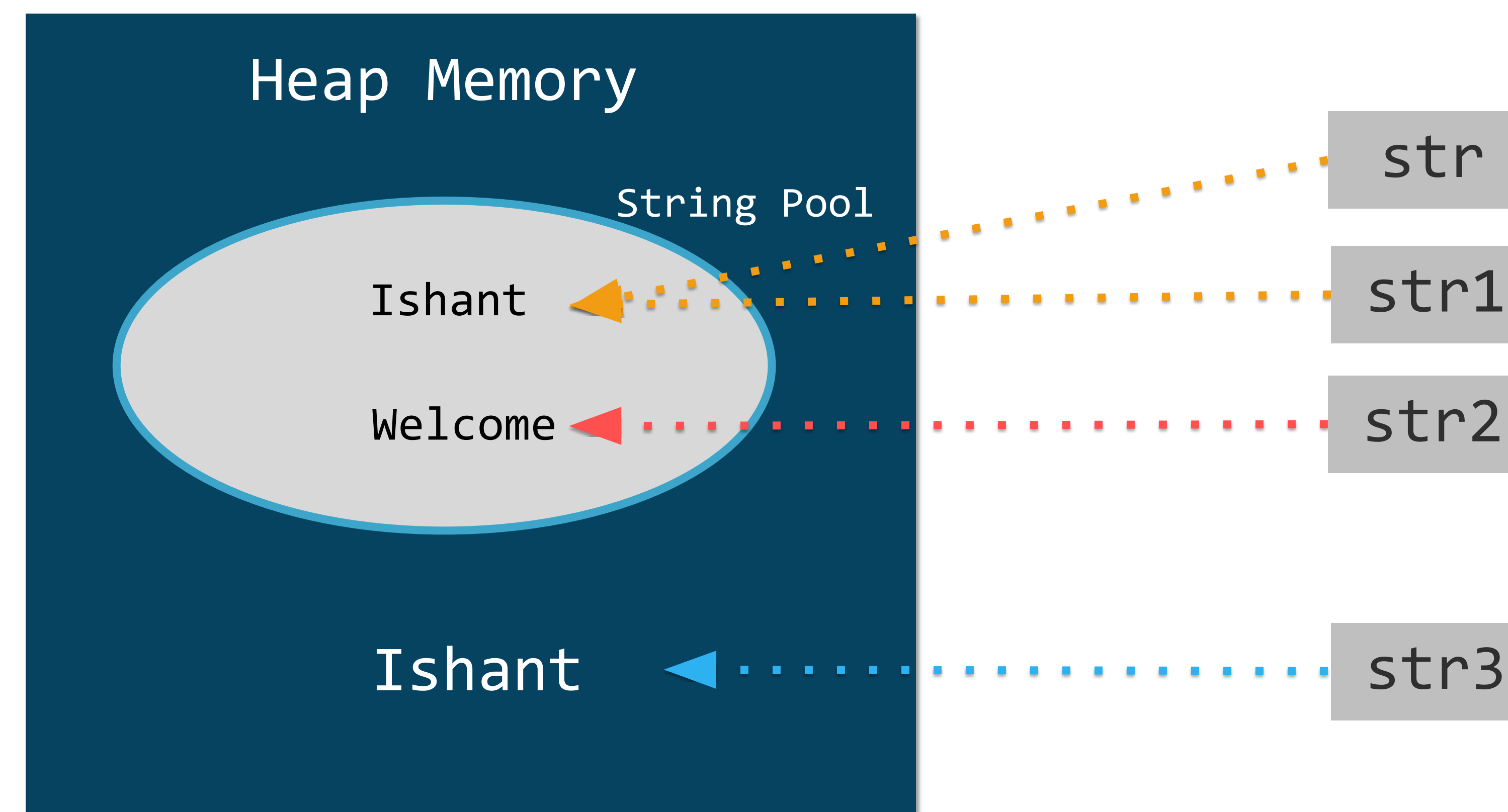




# Creating a String – Using new Keyword

String object created using “new” keyword always creates a new object in heap memory

```
String str = new String (“Ishant”);
```

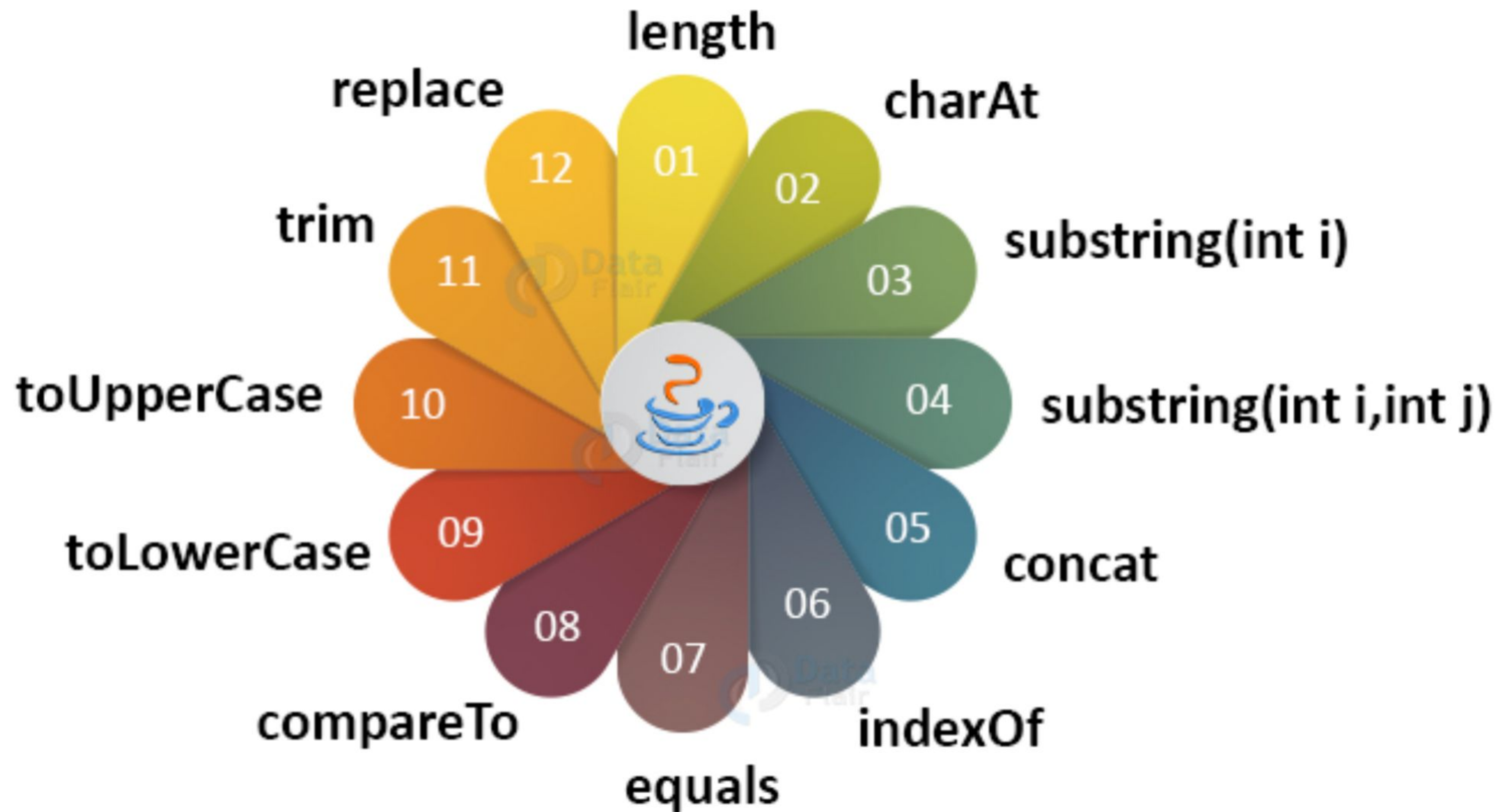




# String Methods



# String Methods





# String Methods

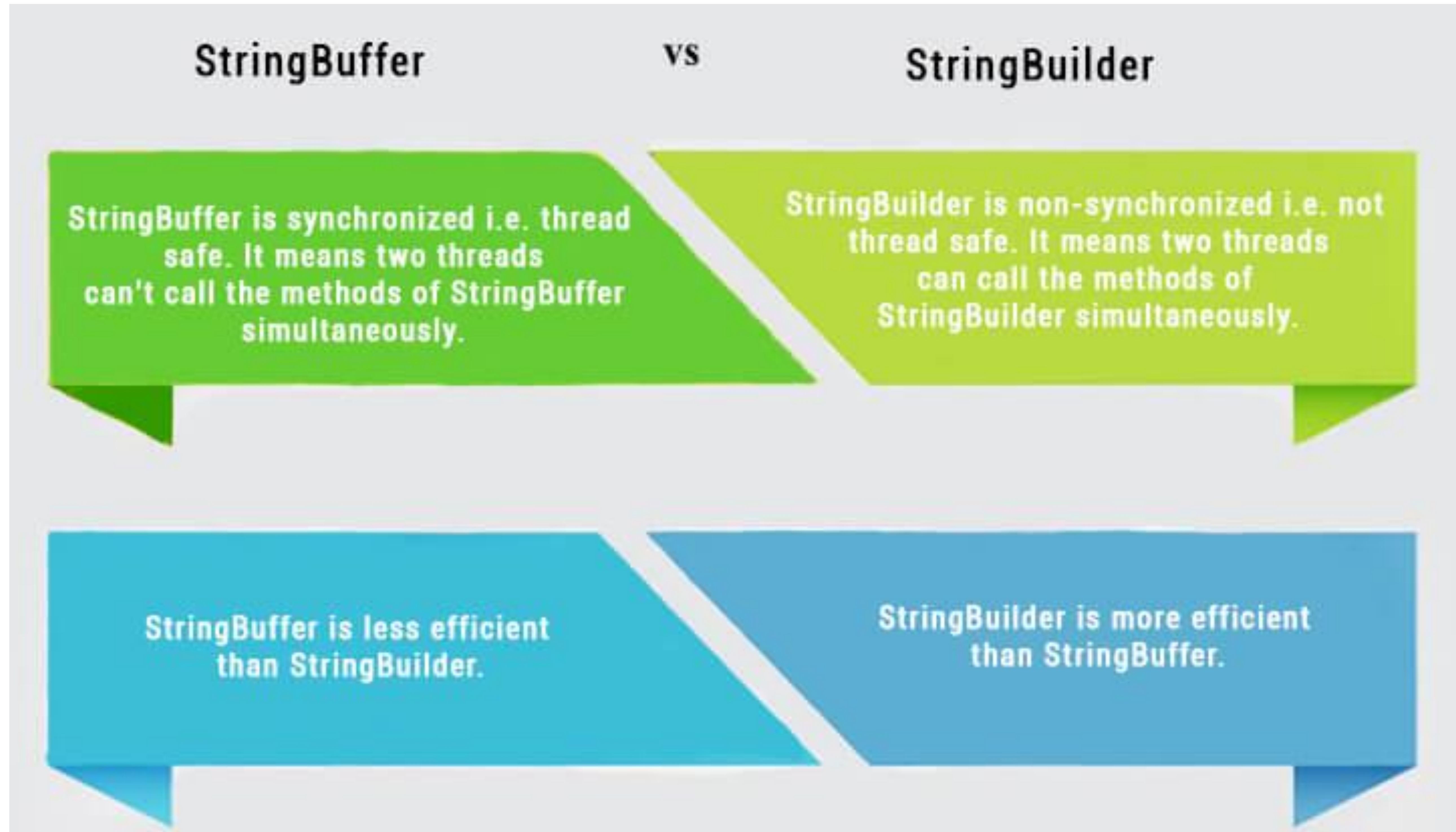
<code>boolean equals(Object another)</code>	Checks the equality of string with the given object
<code>equalsIgnoreCase()</code>	Compares another string without matching the case
<code>length()</code>	Returns a strings length
<code>charAt(i)</code>	Returns a character at a index 'i'
<code>toUpperCase()</code>	Returns the string in uppercase
<code>toLowerCase()</code>	Returns the string in lowercase
<code>replace(oldVal, newVal)</code>	Replaces all occurrences of the specified char value with the given value
<code>trim()</code>	Removes the white spaces from the beginning and ending of string
<code>contains("value")</code>	Checks for the matching sequence of char value and returns true/false
<code>toCharArray()</code>	Converts a string to a new character array
<code>isEmpty()</code>	Checks whether the string is empty or not
<code>endsWith()</code>	Checks if the string ends with the specified suffix
<code>concat()</code>	Concatenates two strings



# StringBuilder and StringBuffer



# StringBuilder vs StringBuffer





# StringTokenizer



**Hello Walking techie how are you**

**StringTokenizer**

**Tokens**

**Hello**

**Walking**

**techie**

**how**

**are**

**you**



**Thank You!**