# JAVA - STRING BUFFER & STRING BUILDER CLASSES

The **StringBuffer** and **StringBuilder** classes are used when there is a necessity to make a lot of modifications to Strings of characters.

Unlike Strings objects of type StringBuffer and Stringbuilder can be modified over and over again with out leaving behind a lot of new unused objects.

The StringBuilder class was introduced as of Java 5 and the main difference between the StringBuffer and StringBuilder is that StringBuilders methods are not thread safe(not Synchronised).

It is recommended to use **StringBuilder** whenever possible because it is faster than StringBuffer. However if thread safety is necessary the best option is StringBuffer objects.

## Example:

```
public class Test{

    public static void main(String args[]){
       StringBuffer sBuffer = new StringBuffer(" test");
       sBuffer.append(" String Buffer");
       System.out.println(sBuffer);
   }
}
```

This would produce the following result:

```
test String Buffer
```

## StringBuffer Methods:

Here is the list of important methods supported by StringBuffer class:

| SN | Methods with Description |
|---|---|
| 1 | public StringBuffer append(String s)<br>Updates the value of the object that invoked the method. The method takes boolean, char, int, long, Strings etc. |
| 2 | public StringBuffer reverse()<br>The method reverses the value of the StringBuffer object that invoked the method. |
| 3 | public delete(int start, int end)<br>Deletes the string starting from start index until end index. |
| 4 | public insert(int offset, int i)<br>This method inserts an string s at the position mentioned by offset. |
| 5 | replace(int start, int end, String str) |

This method replaces the characters in a substring of this StringBuffer with characters in the specified String.

Here is the list of other methods (Except set methods ) which are very similar to String class:

| SN | Methods with Description |
|---|---|
| 1 | **int capacity()**<br>Returns the current capacity of the String buffer. |
| 2 | **char charAt(int index)**<br>The specified character of the sequence currently represented by the string buffer, as indicated by the index argument, is returned. |
| 3 | **void ensureCapacity(int minimumCapacity)**<br>Ensures that the capacity of the buffer is at least equal to the specified minimum. |
| 4 | **void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)**<br>Characters are copied from this string buffer into the destination character array dst. |
| 5 | **int indexOf(String str)**<br>Returns the index within this string of the first occurrence of the specified substring. |
| 6 | **int indexOf(String str, int fromIndex)**<br>Returns the index within this string of the first occurrence of the specified substring, starting at the specified index. |
| 7 | **int lastIndexOf(String str)**<br>Returns the index within this string of the rightmost occurrence of the specified substring. |
| 8 | **int lastIndexOf(String str, int fromIndex)**<br>Returns the index within this string of the last occurrence of the specified substring. |
| 9 | **int length()**<br>Returns the length (character count) of this string buffer. |
| 10 | **void setCharAt(int index, char ch)**<br>The character at the specified index of this string buffer is set to ch. |
| 11 | **void setLength(int newLength)**<br>Sets the length of this String buffer. |
| 12 | **CharSequence subSequence(int start, int end)** |

Returns a new character sequence that is a subsequence of this sequence.

13 **String substring(int start)**

Returns a new String that contains a subsequence of characters currently contained in this StringBuffer.The substring begins at the specified index and extends to the end of the StringBuffer.

14 **String substring(int start, int end)**

Returns a new String that contains a subsequence of characters currently contained in this StringBuffer.

15 **String toString()**

Converts to a string representing the data in this string buffer.