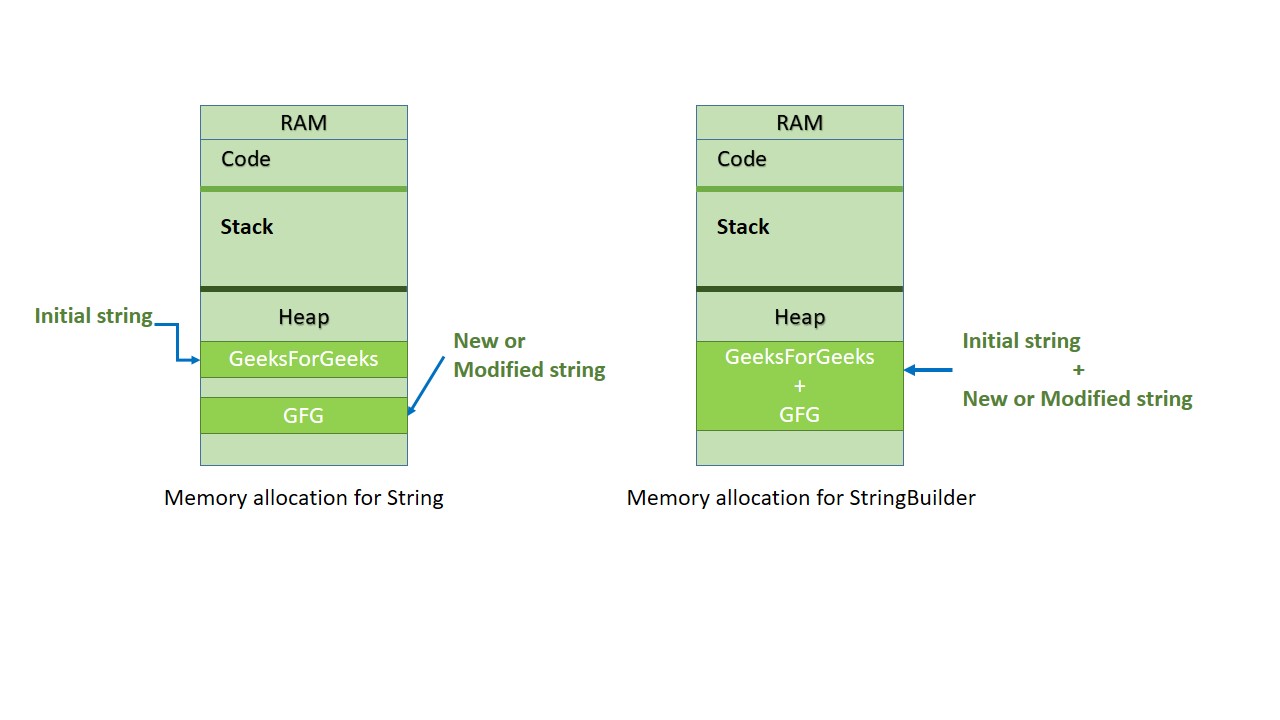
**StringBuilder**

* It represents a mutable string of characters. This class cannot be inherited.
* It means a string cannot be changed once created.
* C# Introduced the **StringBuilder** in the Sytem.Text namespace.
* The **StringBuilder** doesn’t create new object in the memory but dynamically expands memory to accommodate the modified string.

**Namespace:** Using **System.Text;**



**Why we need String Builders?**

To avoid string replacing, appending, removing or inserting new string in the initial string C# introduce **StringBuilder** Object.

**Important Methods:**

* Append(string value)
* AppendLine(string value)
* AppendFormat()
* Insert(int index, string value)
* Remove(int start, int length)
* Replace(old\_val, new|\_val)

**Append:**

This method can be used to add or append a string value of an object to the end of string represented by the current **StringBuilder** object.

**AppendLine:**

This method append the string with a newline at the end.

**AppendFormat:**

It format the input string into the specified format and then append it. This method also appends text to the end of the StringBuilder Object.

**Insert:**

It Inserts the string at specified index in **StringBuilder** object.

**Remove:**

It removes the specified number of characters from the current StringBuilder Object. The removing process beginning at a specified index and extends up to another specified index.

**Replace:**

It replace characters with the **StringBuilder** object with another specified character/string.

**Difference Between String and String Builder:**

|  |  |  |
| --- | --- | --- |
| S.No | String | StringBuilder |
| 1 | Immutable (read only) String | Mutable Instance |
| 2 | Under System Namespace | Under System.Text Namespace |
| 3 | Performance degrades when continuous change of value occurs. | Shows Better Performance Since new Changes are made to an existing instance. |
| 4 | Thread Safe | Not Thread Safe |

