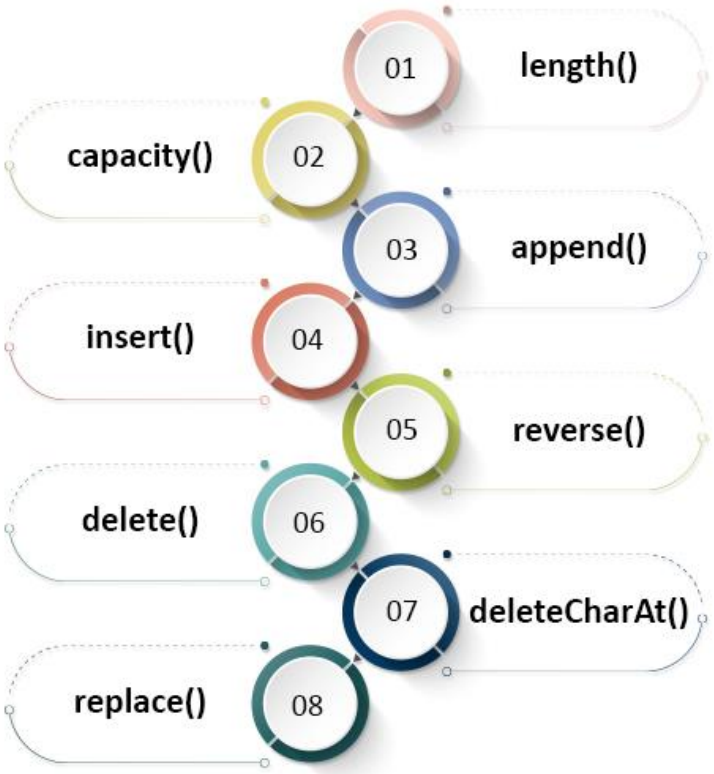
# MÉTODOS DE STRINGBUFFER

Los métodos en Java son la colección de expresiones predefinidas que ejecuta algún tipo de tarea cuando las llamamos explícitamente. Generalmente, para llamar a cualquiera de los métodos, simplemente ejecutamos:

**ReferenceVariable.MethodName (parámetros)**

Ahora sumérjase directamente en los métodos de la clase StringBuffer:




## StringBuffer: length()

Define el número de caracteres en la cadena.

Código para comprender el método length () de la clase StringBuffer:

```
package com.techvidvan.stringbuffer;
import java.io. * ;
class Main {
public static void main(String[] args) {
StringBuffer str = new StringBuffer("ContentWriter");
int len = str.length();
System.out.println("Length : " + len);
}
}
```

Salida:

```
Length: 13
```

## StringBuffer: capacity()

Define la capacidad de la cadena ocupada en el búfer. El método de capacidad también cuenta el espacio reservado ocupado por la cadena.

Código para comprender el método capacity () de la clase StringBuffer:

```
paquete com. techvidvan . stringbuffer ;
importar java. io . * ;
class Main {
public static void main ( String [] args ) {
StringBuffer str = new StringBuffer ( "ContentWriter" ) ;
int cap = str. capacidad () ;
Sistema. fuera . println ( "Capacidad:" + cap ) ;
}
}
```

Salida:

```
Capacity: 29
```

## StringBuffer: append()

El método append () se usa para agregar la cadena o el número al final de la cadena.

```
package com.techvidvan.stringbuffer;
import java.io.*;
class Main
{
public static void main(String[] args)
```

```
{
StringBuffer str = new StringBuffer("Tech");
str.append("Vidvan"); // appends a string in the previously
defined string.
System.out.println(str);
str.append(0); // appends a number in the previously
defined string.
System.out.println(str);
}
}
```

Salida:

```
TechVidvanTechVidvan0
```

## StringBuffer: insert()

El método insert () se utiliza para insertar la cadena en la cadena definida previamente en una posición indexada en particular.

En el método de inserción como parámetro, proporcionamos dos argumentos, el primero es el índice y el segundo es la cadena.

Código para entender el método insert () de la clase StringBuffer:

```
paquete com. techvidvan . stringbuffer ;
public class StringBufferInsert {
public static void main ( String [] args ) {
StringBuffer stringName = new StringBuffer ( "Bienvenido" )
;
Sistema. fuera . println ( stringName ) ;
stringName. insertar ( 8 , "a" ) ;
Sistema. fuera . println ( stringName ) ;
stringName. insertar ( 12 , "TechVidvan" ) ;
Sistema. fuera . println ( stringName ) ;
stringName. insertar ( 22 , "Tutorial" ) ;
Sistema. fuera . println ( stringName ) ;
stringName. insertar ( 31 , "de" ) ;
```

```
Sistema. fuera . println ( stringName ) ;
stringName. insertar ( 35 , "Java" ) ;
Sistema. fuera . println ( stringName ) ;
}
}
```

Salida:

```
Welcome
Welcome to
Welcome to TechVidvan
Welcome to TechVidvan Tutorial
Welcome to TechVidvan Tutorial of
Welcome to TechVidvan Tutorial of Java
```

## StringBuffer: reverse()

El método reverse () invierte todos los caracteres del objeto de la clase StringBuffer. Y, como salida, este método devuelve o da el objeto String invertido.

Código para entender el método reverse () de la clase StringBuffer:

```
package com.techvidvan.stringbuffer;
import java.util. * ;
public class StringBufferReverse {
public static void main(String[] args) {
StringBuffer stringName = new StringBuffer("Welcome to
TechVidvan");
System.out.println("Original String: " + stringName);
stringName.reverse();
System.out.println("Reversed String: " + stringName);
}
}
```

Salida:

```
Original String: Welcome to TechVidvan
Reversed String: navdiVhceT ot emocleW
```

## StringBuffer: delete()

Pasemos al siguiente método de la clase StringBuffer que es el método delete (). El método delete () elimina una secuencia de caracteres del objeto StringBuffer.

Proporcionamos dos argumentos a este método en los que el primer argumento es el índice inicial de la Cadena que queremos eliminar y el segundo índice es el último índice de la Cadena hasta el que queremos eliminar.

Por lo tanto, con este método, podemos eliminar o eliminar una secuencia de caracteres de la cadena. La salida son los personajes. Cadena después de eliminar los caracteres especificados.

Sintaxis:

stringName.delete (int startIndex, int endIndex)

Código para entender el método delete () de la clase StringBuffer:

```
package com.techvidvan.stringbuffer;
import java.io. * ;
public class DeleteMethodDemo {
public static void main(String[] args) {
StringBuffer myString = new StringBuffer("TechVidvanJava");
System.out.println("Original String: " + myString);
myString.delete(0, 4);
System.out.println("Resulting String after deleting
sequence of characters: " + myString);
}
}
```

Saida:

```
Original String: TechVidvanJava
Resulting String after deleting a sequence of characters:
VidvanJava
```

## StringBuffer replace()

El método replace () de la clase StringBuffer reemplaza una secuencia de caracteres con otra secuencia de caracteres.

Toma tres argumentos, uno es el índice inicial y el segundo es el último índice, mientras que el último argumento es la secuencia de caracteres que queremos reemplazar con los caracteres especificados.

Sintaxis:

stringName.replace (int startIndex, int endIndex, String string)

Código para entender el método replace () de la clase StringBuffer:

```
paquete com. techvidvan . stringbuffer ;
importar java. io . * ;
clase pública ReplaceMethodDemo {
public static void main ( String [] args ) {
StringBuffer s = new StringBuffer ( "TechVidvanJava" ) ;
Sistema. fuera . println ( "Cadena original:" + s ) ;
s. reemplazar ( 10 , 14 , "Tutorial" ) ;
Sistema. fuera . println ( "Cadena resultante después de
reemplazar:" + s ) ;
}
}
```

Salida:

```
Cadena original: TechVidvanJava
Cadena resultante después de reemplazar: TechVidvanTutorial
```