**7 Ways to remove spaces**

**1: Using Trim method**

**Trim method ::**

* This method is available from java 1.0.
* By default, this method removes only leading and trailing spaces from a text.
* If this String(current) object represents an empty character sequence, or the first and last characters of character sequence represented by this String object both have codes that are not space (as defined above), then a reference to this String object is returned.
* This method can also work for Unicode i.e Returns a string whose value is this string, with all leading and trailing space removed, where space is defined as any character whose codepoint is less than or equal to 'U+0020' (the space character).
* A String object is returned, representing the substring of this string that begins with the character at index *k* and ends with the character at index *m*-that is, the result of this.substring(k, m +1).
* In Example 4 , we can say that is the whitespaces characters represented by Unicode are available in simultaneously, then trim() eliminates then until the character is identified. If the whitespaces available after those characters are removed to.

**EXAMPLE 1 ::**

In this program we can see , trim() can remove leading spaces(spaces before text) and trailing spaces(spaces after text) not all spaces available in string.

package com.abhi.practice;

public class StringTrimTest {

public static void main(String [] args) {

String s1 = " Hello World !! ";

System.*out*.println(" Before Calling Trim ::" + s1 +"///");

System.*out*.println("After calling Trim ::" +s1.trim()+"///");

}

}

**Output::**

Before Calling Trim :: Hello World !! ///

After calling Trim ::Hello World !!///

**EXAMPLE 2 ::**

From the output we can say that, trim() cannot work with Unicode characters.

**package** com.abhi.practice;

**public** **class** TrimMethodUsingUnicode {

**public** **static** **void** main(String[] args) {

String s1= '\u0020'+"String with space"+ '\u0020';

System.***out***.println("Before Calling Trim ::"+s1);

System.***out***.println("After Calling Trim"+s1.trim());

}

}

**Output::**

Before Calling Trim ::// String with space //

After Calling Trim :://String with space//

**EXAMPLE 3 ::**

Trim method returns the reference of current object.

**package** com.abhi.practice;

**public** **class** StringTrimTest {

**public** **static** **void** main(String [] args) {

String s1="";

// Verifying the hashcode of the object

System.***out***.println("S1 Details ::");

System.***out***.println(System.*identityHashCode*(s1));

System.***out***.println(System.*identityHashCode*(s1.trim()));

System.***out***.println();

String s2 ="java";

System.***out***.println("S2 Details ::");

System.***out***.println(System.*identityHashCode*(s2));

System.***out***.println(System.*identityHashCode*(s2.trim()));

}

}

**Output::**

S1 Details ::

918221580

918221580

S2 Details ::

2055281021

2055281021

**EXAMPLE 4 ::**

We can see from the output that trim method does not work beyond Unicode \u0020.

**Note :: If we write Unicode like \u0027 , \000A , \u000D other which are not allowed to write then we will get compile time error.**

**For eg : String s6=’\u0027’+””;**

**Then we will get compile time error : empty Character Literal**

**package** com.abhi.practice;

**public** **class** TrimMethodUsingUnicode {

**public** **static** **void** main(String[] args) {

String s1 = '\u0001' + "String with space" + '\u0020';

System.***out***.println("Before Calling Trim :://" + s1 + "//");

System.***out***.println("After Calling Trim :://" + s1.trim() + "//");

System.***out***.println();

String s2 = '\u000C' + "" + '\u0009' + "" + '\u0020' + "" + '\u0021' + "" + '\u0020';

System.***out***.println("S2 deatils");

System.***out***.println("//" + s2 + "//");

System.***out***.println("//" + s2.trim() + "//");

String s3 = '\u0001' + "Hello" + '\u0020';

System.***out***.println("S3 details");

System.***out***.println("Before Calling Trim :://" + s3 + "//");

System.***out***.println("After Calling Trim :://" + s3.trim() + "//");

System.***out***.println();

String s4 = '\u0001' + "" + '\u0002' + "" + '\u0003' + "" + '\u0004' + "" + '\u0005' + "" + "" + '\u0006' + ""

+ '\u0007' + "" + '\u0008' + "" + '\u0009' + "" + '\u0010' + "" + "" + '\u0011' + "" + '\u0012' + ""

+ '\u0013' + "" + '\u0014' + "" + '\u0015' + "" + "" + '\u0016' + "" + '\u0017' + "" + '\u0018' + ""

+ '\u0019' + "" + '\u0020';

System.***out***.println("S4 Details");

System.***out***.println("Before Calling Trim :://" + s4 + "//");

System.***out***.println("After Calling Trim :://" + s4.trim() + "//");

System.***out***.println();

String s5 ='\u0021' + "" + '\u0022' + "" + '\u0023' + "" + '\u0024' + "" + '\u0025' + "" + "" + '\u0026' + ""

+ "" + '\u0028' + "" + "" + '\u0029' + "" + "" + '\u0030' + "" + '\u0031' + ""

+ '\u0032' + "" + '\u0033' + "" + '\u0034' + "" + "" + '\u0035' + "" + '\u0036' + "" + '\u0037' + ""

+ '\u0038' + "" + '\u0039'+""+'\u0040';

System.***out***.println("S5 Details");

System.***out***.println("Before Calling Trim :://" + s5 + "//");

System.***out***.println("After Calling Trim :://" + s5.trim() + "//");

}

}

**Output::**

Before Calling Trim :://\_String with space //

After Calling Trim :://String with space//

S2 deatils

//\_ ! //

//!//

S3 details

Before Calling Trim :://\_Hello //

After Calling Trim :://Hello//

S4 Details

Before Calling Trim :://\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ //

After Calling Trim ::////

S5 Details

Before Calling Trim :://!"#$%&()0123456789@//

After Calling Trim :://!"#$%&()0123456789@//

**2 : Using Strip Method**

**Strip method ::**

* This method is available from java 11.
* By default this method removes the leading and trailing spaces from the given string and returns a new string object.
* If this String object represents an empty string, or if all code points in this string are [white space](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Character.html#isWhitespace(int)), then an empty string is returned.
* Otherwise, returns a substring of this string beginning with the first code point that is not a [white space](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Character.html#isWhitespace(int)) up to and including the last code point that is not a [white space](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Character.html#isWhitespace(int)).
* This method internally uses [Character.isWhitespace(int)](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Character.html#isWhitespace(int)) to identify white space characters.

**EXAMPLE 1 ::**

Strip method also removes the leading and trailing spaces only.

**package** com.abhi.practice;

**public** **class** StripMethodUsingSpace {

**public** **static** **void** main(String[] args) {

String s1 =" Hello World !! ";

System.***out***.println("Before Calling Strip ::///"+s1+"///");

System.***out***.println("After Calling Strip ::///"+s1.strip()+"///");

}

}

**Output::**

Before Calling Strip ::/// Hello World !! ///

After Calling Strip ::///Hello World !!///

**EXAMPLE 2 ::**

From the output we can see , strip () can work with Unicode characters.

**package** com.abhi.practice;

**public** **class** StripMethodUsingUnicode {

**public** **static** **void** main(String[] args) {

String s1 = '\u2001'+"String with space"+ '\u2001';

System.***out***.println("Before Calling Strip ::"+s1);

System.***out***.println("After Calling Strip ::"+s1.strip());

}

}

**Output::**

Before Calling Strip ::?String with space?

After Calling Strip ::String with space

**Example 3::**

Combination of example 2 and example 3.

**package** com.abhi.practice;

**public** **class** StripMethodUsingSpace {

**public** **static** **void** main(String[] args) {

String s1 =" Hello World !! ";

System.***out***.println("Before Calling Strip ::///"+s1+"///");

System.***out***.println("After Calling Strip ::///"+s1.strip()+"///");

System.***out***.println();

String s2 ='\u000C'+""+'\u0009'+""+

'\u0020'+""+'\u0021'+""+'\u0020';

System.***out***.println("Checking S2 value:::");

System.***out***.println(s2+"////");

System.***out***.println("Is Blank ::"+s2.strip().isBlank());

System.***out***.println("Is Empty ::"+s2.strip().isEmpty());

System.***out***.println();

System.***out***.println("After Calling Strip method::");

System.***out***.println(s2.strip()+"///");

}

}

**Output::**

Before Calling Strip ::/// Hello World !! ///

After Calling Strip ::///Hello World !!///

Checking S2 value:::

\_ ! ////

Is Blank ::false

Is Empty ::false

After Calling Strip method::

!///

**Example 4 ::**

Using Strip method with empty String and String without space.

**package** com.abhi.practice;

**public** **class** StripMethodWithEmptyStringAndWihoutSpace {

**public** **static** **void** main(String[] args) {

String s1 = "";

System.***out***.println("=".repeat(4) + " S1 Details " + "=".repeat(4));

System.***out***.println("Length of S1 ::"+s1.length());

System.***out***.println("Is Blank ::" + s1.isBlank());

System.***out***.println("Is Empty ::" + s1.isEmpty());

System.***out***.println("S1 object reference before calling strip() ::" + System.*identityHashCode*(s1));

System.***out***.println("S1 object reference After calling strip() ::" + System.*identityHashCode*(s1.strip()));

System.***out***.println();

String s2 = "java";

System.***out***.println("=".repeat(4) + " S1 Details " + "=".repeat(4));

System.***out***.println("Length of S2 ::"+ s2.length());

System.***out***.println("Is Blank ::" + s2.isBlank());

System.***out***.println("Is Empty ::" + s2.isEmpty());

System.***out***.println("S1 object reference before calling strip() ::" + System.*identityHashCode*(s2));

System.***out***.println("S1 object reference After calling strip() ::" + System.*identityHashCode*(s2.strip()));

}

}

**Output::**

==== S1 Details ====

Length of S1 ::0

Is Blank ::true

Is Empty ::true

S1 object reference before calling strip() ::918221580

S1 object reference After calling strip() ::918221580

==== S1 Details ====

Length of S2 ::4

Is Blank ::false

Is Empty ::false

S1 object reference before calling strip() ::2055281021

S1 object reference After calling strip() ::2055281021

**Example 5 ::**

**package** com.abhi.practice;

**public** **class** TrimMethodUsingUnicode {

**public** **static** **void** main(String[] args) {

String s1 = '\u0001' + "String with space" + '\u0020';

System.***out***.println("=".repeat(4) + " S1 Details " + "=".repeat(4));

System.***out***.println("Before Calling Trim :://" + s1 + "//");

System.***out***.println("After Calling Trim :://" + s1.trim() + "//");

System.***out***.println();

String s2 = '\u000C' + "" + '\u0009' + "" + '\u0020' + "" + '\u0021' + "" + '\u0020';

System.***out***.println("=".repeat(4) + " S2 Details " + "=".repeat(4));

System.***out***.println("//" + s2 + "//");

System.***out***.println("//" + s2.trim() + "//");

System.***out***.println();

String s3 = '\u0001' + "Hello" + '\u0020';

System.***out***.println("=".repeat(4)+" S3 Details "+"=".repeat(4));

System.***out***.println("Before Calling Trim :://" + s3 + "//");

System.***out***.println("After Calling Trim :://" + s3.trim() + "//");

System.***out***.println();

String s4 = '\u0001' + "" + '\u0002' + "" + '\u0003' + "" + '\u0004' + "" + '\u0005' + "" + "" + '\u0006' + ""

+ '\u0007' + "" + '\u0008' + "" + '\u0009' + "" + '\u0010' + "" + "" + '\u0011' + "" + '\u0012' + ""

+ '\u0013' + "" + '\u0014' + "" + '\u0015' + "" + "" + '\u0016' + "" + '\u0017' + "" + '\u0018' + ""

+ '\u0019' + "" + '\u0020';

System.***out***.println("=".repeat(4)+" S4 Details "+"=".repeat(4));

System.***out***.println("Before Calling Trim :://" + s4 + "//");

System.***out***.println("After Calling Trim :://" + s4.trim() + "//");

System.***out***.println();

String s5 = '\u0021' + "" + '\u0022' + "" + '\u0023' + "" + '\u0024' + "" + '\u0025' + "" + "" + '\u0026' + ""

+ "" + '\u0028' + "" + "" + '\u0029' + "" + "" + '\u0030' + "" + '\u0031' + "" + '\u0032' + ""

+ '\u0033' + "" + '\u0034' + "" + "" + '\u0035' + "" + '\u0036' + "" + '\u0037' + "" + '\u0038' + ""

+ '\u0039' + "" + '\u0040';

System.***out***.println("=".repeat(4)+" S5 Details "+"=".repeat(4));

System.***out***.println("Before Calling Trim :://" + s5 + "//");

System.***out***.println("After Calling Trim :://" + s5.trim() + "//");

}

}

**Output::**

==== S1 Details ====

Before Calling Trim :://\_String with space //

After Calling Trim :://String with space//

==== S2 Details ====

//\_ ! //

//!//

==== S3 Details ====

Before Calling Trim :://\_Hello //

After Calling Trim :://Hello//

==== S4 Details ====

Before Calling Trim :://\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ //

After Calling Trim ::////

==== S5 Details ====

Before Calling Trim :://!"#$%&()0123456789@//

After Calling Trim :://!"#$%&()0123456789@//

**StripLeading Method**

* This method was introduced in java 11.
* It removes all the leading whitespaces and returns a String.
* If current String is empty, then it returns the current object reference.
* If String contains only trailing spaces, then the current object reference is returned.
* If all code points in this string are [white space](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Character.html#isWhitespace(int)), then an empty string is returned.
* This method is used to remove white spaces from the beginning of a String.
* We cannot create empty String using Unicode .
* This method internally uses [Character.isWhitespace(int)](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Character.html#isWhitespace(int)).

**Example 1:**

**package** com.abhi.practice;

**public** **class** StripLeadingMethodWithUsingSpace {

**public** **static** **void** main(String [] args) {

String s1 = " !Hello World";

System.***out***.println("=".repeat(4)+"S1 Details"+"=".repeat(4));

System.***out***.println("Before Calling StripLeading ::"+s1);

System.***out***.println("After Calling StripLeading ::"+s1.stripLeading());

}// main(-)

}// class

**Output::**

====S1 Details====

Before Calling StripLeading :: !Hello World

After Calling StripLeading ::!Hello World

**Example 2::**

When String contains trailing spaces.

**package** com.abhi.practice;

**public** **class** StripLeadingMethodWithUsingSpace {

**public** **static** **void** main(String [] args) {

String s2 = "!Hello World ";

System.***out***.println("=".repeat(4)+"S2 Details"+"=".repeat(4));

System.***out***.println("Before Calling StripLeading ::"+s2+"///");

System.***out***.println("After Calling StripLeading ::"+s2.stripLeading()+"///");

}// main(-)

}// class

**Output ::**

====S2 Details====

Before Calling StripLeading ::!Hello World ///

After Calling StripLeading ::!Hello World ///

**Example 3::**

Verifying the reference of return object after calling and before calling StripLeading().

**package** com.abhi.practice;

**public** **class** StripLeadingMethodWithUsingSpace {

**public** **static** **void** main(String [] args) {

String s1 = " !Hello World";

System.***out***.println("=".repeat(4)+"S1 Details"+"=".repeat(4));

System.***out***.println("Before Calling StripLeading ::"+s1);

System.***out***.println("S1 object reference before calling method ::"+System.*identityHashCode*(s1));

System.***out***.println("After Calling StripLeading ::"+s1.stripLeading());

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s1.stripLeading()));

System.***out***.println("S1 object reference after calling method ::"+System.*identityHashCode*(s1));

System.***out***.println("=".repeat(15));

System.***out***.println();

String s2 = "!Hello World ";

System.***out***.println("=".repeat(4)+"S2 Details"+"=".repeat(4));

System.***out***.println("Before Calling StripLeading ::"+s2);

System.***out***.println("S2 object reference before calling method ::"+System.*identityHashCode*(s2));

System.***out***.println("After Calling StripLeading ::"+s2.stripLeading());

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s2.stripLeading()));

System.***out***.println("S2 object reference after calling method ::"+System.*identityHashCode*(s2));

System.***out***.println("=".repeat(15));

}// main(-)

}// class

**Output ::**

====S1 Details====

Before Calling StripLeading :: !Hello World

S1 object reference before calling method ::918221580

After Calling StripLeading ::!Hello World

Returned object reference after calling method ::2055281021

S1 object reference after calling method ::918221580

===============

====S2 Details====

Before Calling StripLeading ::!Hello World

S2 object reference before calling method ::1554547125

After Calling StripLeading ::!Hello World

Returned object reference after calling method ::1554547125

S2 object reference after calling method ::1554547125

===============

**Example 4::**

Working with empty String.

**package** com.abhi.practice;

**public** **class** StripLeadingMethodUsingEmptyString {

**public** **static** **void** main(String [] args) {

String s3 ="";

System.***out***.println("Verifying String is empty or not before calling method");

System.***out***.println("Is Empty ::"+s3.isEmpty());

System.***out***.println("Is Blank ::"+s3.isBlank());

System.***out***.println("Length ::"+s3.length());

System.***out***.println();

System.***out***.println("=".repeat(4)+"S3 Details"+"=".repeat(4));

System.***out***.println("S3 value before calling method ::"+s3);

System.***out***.println("S3 object reference before callling method ::"+System.*identityHashCode*(s3));

System.***out***.println("Returned object reference after callling method ::"+System.*identityHashCode*(s3.stripLeading()));

System.***out***.println("S3 value after calling method ::"+s3);

System.***out***.println("S3 object reference after callling method ::"+System.*identityHashCode*(s3));

}// main(-)

}// class

**Output ::**

Verifying String is empty or not before calling method

Is Empty ::true

Is Blank ::true

Length ::0

====S3 Details====

S3 value before calling method ::

S3 object reference before callling method ::918221580

Returned object reference after callling method ::918221580

S3 value after calling method ::

S3 object reference after callling method ::918221580

**Example 6::**

Removing spaces in the form of Unicode .

**package** com.abhi.unicode;

**public** **class** StripLeadingMethodUsingUniodeAtLeading {

**public** **static** **void** main(String[] args) {

String s1 = "\u0020\u0020\u0020!Hello World";

System.***out***.println("=".repeat(4)+"S1 Details"+"=".repeat(4));

System.***out***.println("S1 value before calling method ::"+s1);

System.***out***.println("Returned Object value after caling method ::"+s1.stripLeading());

}// main(-)

}// class

**Output ::**

====S1 Details====

S1 value before calling method :: !Hello World

Returned Object value after caling method ::!Hello World

**Example 7::**

**package** com.abhi.unicode;

**public** **class** StripMethodUsingUnicodeAtTrailing {

**public** **static** **void** main(String[] args) {

String s1 = "!Hello World"+'\u0020'+'\u0020'+'\u0020';

System.***out***.println("=".repeat(4)+"S1 Details"+"=".repeat(4));

System.***out***.println("S1 value before calling method ::"+s1+"///");

System.***out***.println("Returned Object value after calling method ::"+s1.stripLeading()+"///");

}// main(-)

}// class

**Output ::**

====S1 Details====

S1 value before calling method ::!Hello World ///

Returned Object value after calling method ::!Hello World ///

**Example 8::**

We cannot create empty String using Unicode.

**package** com.abhi.unicode;

**public** **class** StripLeadingMethodForEmptyUnicode {

**public** **static** **void** main(String[]args) {

String s1 ="\u0000";

System.***out***.println("S1 Value ::"+s1);

System.***out***.println("Verifying wheather S1 is empty or not");

System.***out***.println("Is Empty ::"+s1.isEmpty());

System.***out***.println("Is Blank ::"+s1.isBlank());

System.***out***.println("Length ::"+s1.length());

System.***out***.println("\*".repeat(40));

String s2 ='\u0000'+"";

System.***out***.println("S2 Value ::"+s2);

System.***out***.println("Verifying wheather S2 is empty or not");

System.***out***.println("Is Empty ::"+s2.isEmpty());

System.***out***.println("Is Blank ::"+s2.isBlank());

System.***out***.println("Length ::"+s2.length());

System.***out***.println();

System.***out***.println("=".repeat(20)+"Checking S1 Object References"+"=".repeat(20));

System.***out***.println("S1 object reference before calling StripLeading ::"+System.*identityHashCode*(s1));

System.***out***.println("Returned Object Reference On calling StripLeading ::"+System.*identityHashCode*(s1.stripLeading()));

System.***out***.println("S1 object reference before calling StripLeading ::"+System.*identityHashCode*(s1));

System.***out***.println();

System.***out***.println("=".repeat(20)+"Checking S2 Object References"+"=".repeat(20));

System.***out***.println("S1 object reference before calling StripLeading ::"+System.*identityHashCode*(s2));

System.***out***.println("Returned Object Reference On calling StripLeading ::"+System.*identityHashCode*(s2.stripLeading()));

System.***out***.println("S1 object reference before calling StripLeading ::"+System.*identityHashCode*(s2));

}// main(-)

}// class

**Output ::**

S1 Value ::\_

Verifying whether S1 is empty or not

Is Empty ::false

Is Blank ::false

Length ::1

S2 Value ::\_

Verifying wheather S2 is empty or not

Is Empty ::false

Is Blank ::false

Length ::1

====================Checking S1 Object References====================

S1 object reference before calling StripLeading ::918221580

Returned Object Reference On calling StripLeading ::918221580

S1 object reference before calling StripLeading ::918221580

====================Checking S2 Object References====================

S1 object reference before calling StripLeading ::918221580

Returned Object Reference On calling StripLeading ::918221580

S1 object reference before calling StripLeading ::918221580

**Example 9::**

Verifying Object references before and after calling stripLeading().

**package** com.abhi.unicode;

**public** **class** VerifyingObjectReferencesBeforeAndAfterCallingStripLeading {

**public** **static** **void** main(String[] args) {

String s1 = "\u0020\u0020\u0020!Hello World";

System.***out***.println("=".repeat(10)+"S1 Details"+"=".repeat(10));

System.***out***.println("S1 value before calling method ::"+s1);

System.***out***.println("S1 Object reference before calling method ::"+System.*identityHashCode*(s1));

System.***out***.println("Returned Object value after calling method ::"+s1.stripLeading());

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s1.stripLeading()));

System.***out***.println("S1 value after calling method ::"+s1);

System.***out***.println("S1 Object reference after calling method ::"+System.*identityHashCode*(s1));

System.***out***.println();

String s2 ="!Hello World\u0020\u0020\u0020";

System.***out***.println("=".repeat(10)+"S2 Details"+"=".repeat(10));

System.***out***.println("S2 value before calling method ::"+s2+"///");

System.***out***.println("S2 Object reference before calling method ::"+System.*identityHashCode*(s2));

System.***out***.println("Returned Object value after calling method ::"+s2.stripLeading()+"///");

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s2.stripLeading()));

System.***out***.println("S2 value after calling method ::"+s2+"///");

System.***out***.println("S2 Object reference after calling method ::"+System.*identityHashCode*(s2));

}//main(-)

}// class

**Output ::**

==========S1 Details==========

S1 value before calling method :: !Hello World

S1 Object reference before calling method ::918221580

Returned Object value after calling method ::!Hello World

Returned object reference after calling method ::2055281021

S1 value after calling method :: !Hello World

S1 Object reference after calling method ::918221580

==========S2 Details==========

S2 value before calling method ::!Hello World ///

S2 Object reference before calling method ::1554547125

Returned Object value after caling method ::!Hello World ///

Returned object reference after calling method ::1554547125

S2 value after calling method ::!Hello World ///

S2 Object reference after calling method ::1554547125

**Example 10::**

When there are no leading spaces then StripLeading method returns the current object reference.

**package** com.abhi.practice;

**public** **class** StripMethodWithoutHavingLeadingOrTrailingSpaces {

**public** **static** **void** main(String[] args) {

String s1 = "!Hello World";

System.***out***.println("=".repeat(10)+"S1 Details"+"=".repeat(10));

System.***out***.println("S1 value before calling method ::"+s1);

System.***out***.println("S1 Object reference before calling method ::"+System.*identityHashCode*(s1));

System.***out***.println("Returned Object value after calling method ::"+s1.stripLeading());

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s1.stripLeading()));

System.***out***.println("S1 value after calling method ::"+s1);

System.***out***.println("S1 Object reference after calling method ::"+System.*identityHashCode*(s1));

System.***out***.println();

}// main(-)

}// class

**Output ::**

==========S1 Details==========

S1 value before calling method ::!Hello World

S1 Object reference before calling method ::918221580

Returned Object value after calling method ::!Hello World

Returned object reference after calling method ::918221580

S1 value after calling method ::!Hello World

S1 Object reference after calling method ::918221580

**Strip Trailing Method ::**

* This method was introduced in java 11.
* It removes all the white spaces and returns a String object.
* If current String is empty then it returns the current object reference.
* If String contains only leading Spaces then it returns current object reference.
* If all characters in this string are [white space](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Character.html#isWhitespace(int)), then an empty string is returned.
* This method is used to remove white spaces from the end of a String.
* We cannot create empty String using Unicode .
* This method internally uses [Character.isWhitespace(int)](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/lang/Character.html#isWhitespace(int)).

**Example 1::**

**package** com.abhi.practice;

**public** **class** StripTrailingTrailingMethodUsingSpace {

**public** **static** **void** main(String[] args) {

String s1 ="Hello World !! ";

System.***out***.println("Before Calling StripTrailing ::"+s1+"///");

System.***out***.println("After Calling StripTrailing ::"+s1.stripTrailing()+"///");

System.***out***.println();

String s2 ='\u000C'+""+'\u0009'+""+ '\u0020'+""+'\u0021'+""+'\u0020';

System.***out***.println("Checking S2 value:::");

System.***out***.println("////"+s2);

System.***out***.println("Is Blank ::"+s2.stripTrailing().isBlank());

System.***out***.println("Is Empty ::"+s2.stripTrailing().isEmpty());

System.***out***.println();

System.***out***.println("After Calling StripTrailing method::");

System.***out***.println(s2.stripTrailing()+"///");

}

}

**Output ::**

Before Calling StripTrailing ::Hello World !! ///

After Calling StripTrailing ::Hello World !!///

Checking S2 value:::

////\_ !

Is Blank ::false

Is Empty ::false

After Calling StripTrailing method::

\_ !///

**Example 2::**

**package** com.abhi.practice;

**public** **class** CheckingObjectReferenceAfterCallingStripTrailingMethod {

**public** **static** **void** main(String [] args) {

String s1 = "!Hello World ";

System.***out***.println("=".repeat(10)+"S1 Details"+"=".repeat(10));

System.***out***.println("S1 value Before Calling StripTrailing ::"+s1+"///");

System.***out***.println("S1 object reference before calling method ::"+System.*identityHashCode*(s1));

System.***out***.println("S1 Value After Calling StripTrailing ::"+s1.stripTrailing()+"///");

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s1.stripTrailing()));

System.***out***.println("S1 object reference after calling method ::"+System.*identityHashCode*(s1));

System.***out***.println("=".repeat(15));

System.***out***.println();

String s2 = " !Hello World";

System.***out***.println("=".repeat(10)+"S2 Details"+"=".repeat(10));

System.***out***.println("S2 value Before Calling StripTrailing ::"+s2+"///");

System.***out***.println("S2 object reference before calling method ::"+System.*identityHashCode*(s2));

System.***out***.println("S2 Value After Calling StripTrailing ::"+s2.stripTrailing()+"///");

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s2.stripTrailing()));

System.***out***.println("S2 object reference after calling method ::"+System.*identityHashCode*(s2));

System.***out***.println("=".repeat(15));

}// main(-)

}//class

**Output ::**

==========S1 Details==========

S1 value Before Calling StripTrailing ::!Hello World ///

S1 object reference before calling method ::918221580

S1 Value After Calling StripTrailing ::!Hello World///

Returned object reference after calling method ::2055281021

S1 object reference after calling method ::918221580

===============

==========S2 Details==========

S2 value Before Calling StripTrailing :: !Hello World///

S2 object reference before calling method ::1554547125

S2 Value After Calling StripTrailing :: !Hello World///

Returned object reference after calling method ::1554547125

S2 object reference after calling method ::1554547125

===============

**Example 3::**

**package** com.abhi.practice;

**public** **class** StripTrailingUsingEmptyString {

**public** **static** **void** main(String [] args) {

String s3 ="";

System.***out***.println("Verifying String is empty or not before calling method");

System.***out***.println("Is Empty ::"+s3.isEmpty());

System.***out***.println("Is Blank ::"+s3.isBlank());

System.***out***.println("Length ::"+s3.length());

System.***out***.println();

System.***out***.println("=".repeat(4)+"S3 Details"+"=".repeat(4));

System.***out***.println("S3 value before calling method ::"+s3);

System.***out***.println("S3 object reference before calling method ::"+System.*identityHashCode*(s3));

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s3.stripTrailing()));

System.***out***.println("S3 value after calling method ::"+s3);

System.***out***.println("S3 object reference after calling method ::"+System.*identityHashCode*(s3));

}// main(-)

}

**Output ::**

Verifying String is empty or not before calling method

Is Empty ::true

Is Blank ::true

Length ::0

====S3 Details====

S3 value before calling method ::

S3 object reference before calling method ::918221580

Returned object reference after calling method ::918221580

S3 value after calling method ::

S3 object reference after calling method ::918221580

**Example 4::**

**package** com.abhi.practice;

**public** **class** StripTrailingMethodWithoutSpacesInString {

**public** **static** **void** main(String[] args) {

String s4="Java";

System.***out***.println("=".repeat(10)+"S4 Details"+"=".repeat(10));

System.***out***.println("S4 value Before Calling StripTrailing ::"+s4+"///");

System.***out***.println("S4 object reference before calling method ::"+System.*identityHashCode*(s4));

System.***out***.println("Returned Object Value After Calling StripTrailing ::"+s4.stripTrailing()+"///");

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s4.stripTrailing()));

System.***out***.println("S4 value After Calling StripTrailing ::"+s4+"///");

System.***out***.println("S4 object reference after calling method ::"+System.*identityHashCode*(s4));

System.***out***.println("=".repeat(15));

}// main(-)

}// class

**Output ::**

==========S4 Details==========

S4 value Before Calling StripTrailing ::Java///

S4 object reference before calling method ::918221580

Returned Object Value After Calling StripTrailing ::Java///

Returned object reference after calling method ::918221580

S4 value After Calling StripTrailing ::Java///

S4 object reference after calling method ::918221580

===============

**Example 5::**

**package** com.abhi.unicode;

**public** **class** StripTrailingRemovingUnicodeAsSpacesFromString{

**public** **static** **void** main(String [] args) {

String s1 = "\u0020\u0020\u0020!Hello World";

System.***out***.println("=".repeat(10)+"S1 Details"+"=".repeat(10));

System.***out***.println("S1 value before calling method ::"+s1);

System.***out***.println("Returned Object value after calling method ::"+s1.stripTrailing());

System.***out***.println();

String s2 = "!Hello World"+'\u0020'+'\u0020'+'\u0020';

System.***out***.println("=".repeat(10)+"S2 Details"+"=".repeat(10));

System.***out***.println("S1 value before calling method ::"+s2+"///");

System.***out***.println("Returned Object value after calling method ::"+s2.stripTrailing()+"///");

}// main(-)

}// class

**Output ::**

==========S1 Details==========

S1 value before calling method :: !Hello World

Returned Object value after calling method :: !Hello World

==========S2 Details==========

S1 value before calling method ::!Hello World ///

Returned Object value after calling method ::!Hello World///

**Example 6::**

**package** com.abhi.unicode;

**public** **class** VerifyingObjectReferenceBeforeAndAfterCallingStripTrailing {

**public** **static** **void** main(String[] args) {

String s1 = "\u0020\u0020\u0020!Hello World";

System.***out***.println("=".repeat(10)+"S1 Details"+"=".repeat(10));

System.***out***.println("S1 value before calling method ::"+s1);

System.***out***.println("S1 Object reference before calling method ::"+System.*identityHashCode*(s1));

System.***out***.println("Returned Object value after caling method ::"+s1.stripTrailing());

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s1.stripTrailing()));

System.***out***.println("S1 value after calling method ::"+s1);

System.***out***.println("S1 Object reference after calling method ::"+System.*identityHashCode*(s1));

System.***out***.println();

String s2 ="!Hello World\u0020\u0020\u0020";

System.***out***.println("=".repeat(10)+"S2 Details"+"=".repeat(10));

System.***out***.println("S2 value before calling method ::"+s2+"///");

System.***out***.println("S2 Object reference before calling method ::"+System.*identityHashCode*(s2));

System.***out***.println("Returned Object value after caling method ::"+s2.stripTrailing()+"///");

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s2.stripTrailing()));

System.***out***.println("S2 value after calling method ::"+s2+"///");

System.***out***.println("S2 Object reference after calling method ::"+System.*identityHashCode*(s2));

}//main(-)

}// class

**Output ::**

==========S1 Details==========

S1 value before calling method :: !Hello World

S1 Object reference before calling method ::918221580

Returned Object value after calling method :: !Hello World

Returned object reference after calling method ::918221580

S1 value after calling method :: !Hello World

S1 Object reference after calling method ::918221580

==========S2 Details==========

S2 value before calling method ::!Hello World ///

S2 Object reference before calling method ::2055281021

Returned Object value after calling method ::!Hello World///

Returned object reference after calling method ::1554547125

S2 value after calling method ::!Hello World ///

S2 Object reference after calling method ::2055281021

**Replace Method(char oldChar, char newChar) ::**

* This method was introduced in jdk 1.5.
* This method replaces all the occurrences of oldChar with newchar.
* If the character oldChar does not occur in the character sequence represented by this String object, then a reference to this String object is returned.
* Replace method can remove space(\u0020) from the given String.

**Example 1 ::**

**package** com.abhi.practice;

**public** **class** RemovingSpaceUsingReplaceMethod {

**public** **static** **void** main(String[] args) {

String s1 =" !Hello World ";

System.***out***.println("=".repeat(10)+"S1 object Details"+"=".repeat(10));

System.***out***.println("S1 object value before calling method ::"+s1+"///");

System.***out***.println("S1 Object reference before calling method ::"+System.*identityHashCode*(s1));

System.***out***.println("Returned object value after calling method ::"+s1.replace(" ", "")+"///");

System.***out***.println("Returned Object reference after calling method ::"+System.*identityHashCode*(s1.replace(" ", "")));

System.***out***.println("S1 object value after calling method ::"+s1+"///");

System.***out***.println("S1 Object reference after calling method ::"+System.*identityHashCode*(s1));

}// main(-)

}// class

**Output ::**

==========S1 object Details==========

S1 object value before calling method :: !Hello World ///

S1 Object reference before calling method ::918221580

Returned object value after calling method ::!HelloWorld///

Returned Object reference after calling method ::2055281021

S1 object value after calling method :: !Hello World ///

S1 Object reference after calling method ::918221580

**Example 2 ::**

When the given char to be replaced is not found, then the current object reference is returned.

**package** com.abhi.practice;

**public** **class** ReplaceMethodWhenGivenCharIsNotFound {

**public** **static** **void** main(String[] args) {

String s1 ="!HelloWorld";

System.***out***.println("=".repeat(10)+"S1 object Details"+"=".repeat(10));

System.***out***.println("S1 object value before calling method ::"+s1+"///");

System.***out***.println("S1 Object reference before calling method ::"+System.*identityHashCode*(s1));

System.***out***.println("Returned object value after calling method ::"+s1.replace(" ", "")+"///");

System.***out***.println("Returned Object reference after calling method ::"+System.*identityHashCode*(s1.replace(" ", "")));

System.***out***.println("S1 object value after calling method ::"+s1+"///");

System.***out***.println("S1 Object reference after calling method ::"+System.*identityHashCode*(s1));

}// main(-)

}//class

**Output ::**

==========S1 object Details==========

S1 object value before calling method ::!HelloWorld///

S1 Object reference before calling method ::918221580

Returned object value after calling method ::!HelloWorld///

Returned Object reference after calling method ::918221580

S1 object value after calling method ::!HelloWorld///

S1 Object reference after calling method ::918221580

**Example 3 ::**

**package** com.abhi.unicode;

**public** **class** ReplaceMethodUsingUnicode {

**public** **static** **void** main(String[] args) {

String s1 = "\u0020\u0020\u0020!Hello World";

System.***out***.println("=".repeat(10)+"S1 Details"+"=".repeat(10));

System.***out***.println("S1 value before calling method ::"+s1);

System.***out***.println("S1 Object reference before calling method ::"+System.*identityHashCode*(s1));

System.***out***.println("Returned Object value after caling method ::"+s1.replace(" ", ""));

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s1.replace(" ", "")));

System.***out***.println("S1 value after calling method ::"+s1);

System.***out***.println("S1 Object reference after calling method ::"+System.*identityHashCode*(s1));

System.***out***.println();

String s2 ="!Hello World\u0020\u0020\u0020";

System.***out***.println("=".repeat(10)+"S2 Details"+"=".repeat(10));

System.***out***.println("S2 value before calling method ::"+s2+"///");

System.***out***.println("S2 Object reference before calling method ::"+System.*identityHashCode*(s2));

System.***out***.println("Returned Object value after caling method ::"+s2.replace(" ", "")+"///");

System.***out***.println("Returned object reference after calling method ::"+System.*identityHashCode*(s2.replace(" ", "")));

System.***out***.println("S2 value after calling method ::"+s2+"///");

System.***out***.println("S2 Object reference after calling method ::"+System.*identityHashCode*(s2));

}//main(-)

}// class

**Output ::**

==========S1 Details==========

S1 value before calling method :: !Hello World

S1 Object reference before calling method ::918221580

Returned Object value after caling method ::!HelloWorld

Returned object reference after calling method ::2055281021

S1 value after calling method :: !Hello World

S1 Object reference after calling method ::918221580

==========S2 Details==========

S2 value before calling method ::!Hello World ///

S2 Object reference before calling method ::1554547125

Returned Object value after caling method ::!HelloWorld///

Returned object reference after calling method ::617901222

S2 value after calling method ::!Hello World ///

S2 Object reference after calling method ::1554547125

**ReplaceAll Method(String regex , String replacement)**

* This method was introduced in jdk1.4.
* This method can work with regular expression.
* Replaces each substring of this string that matches the given [regular expression](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/regex/Pattern.html#sum) with the given replacement.

**Note** :: That backslashes (\) and dollar signs ($) in the replacement string may cause the results to be different than if it were being treated as a literal replacement string; see [Matcher.replaceAll](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/regex/Matcher.html" \l "replaceAll(java.lang.String)). Use [Matcher.quoteReplacement(java.lang.String)](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/regex/Matcher.html" \l "quoteReplacement(java.lang.String)) to suppress the special meaning of these characters, if desired.

**Example 1 ::**

We just need to create correct regular expression with correct replacement parameter. Some regular expression examples as below:

|  |  |
| --- | --- |
| \s+ | Find all space |
| ^\s+ | Find all spaces at line beginning |
| \s+$ | Find all spaces at line ending |

**package** com.abhi.practice;

**public** **class** ReplaceAllTest {

**public** **static** **void** main(String[] args) {

String string = " String with space ";

System.***out***.println("Before replaceAll : \"" + string+"\"");

System.***out***.println("String object reference before replaceAll ::"+System.*identityHashCode*(string));

System.***out***.println();

System.***out***.println("Replace all space : \"" + string.replaceAll(" ", "") + "\"");

System.***out***.println("Returned Object reference ::"+System.*identityHashCode*(string.replaceAll(" ", "") + "\""));

System.***out***.println();

System.***out***.println("Replace all regex : \"" + string.replaceAll("\\s+", "") + "\"");

System.***out***.println("Returned Object reference ::"+System.*identityHashCode*(string.replaceAll("\\s+", "") + "\""));

System.***out***.println();

System.***out***.println("Replace Leading : \"" + string.replaceAll("^\\s+", "") + "\"");

System.***out***.println("Returned Object reference ::"+System.*identityHashCode*(string.replaceAll("^\\s+", "") + "\""));

System.***out***.println();

System.***out***.println("Replace trailing : \"" + string.replaceAll("\\s+$", "") + "\"");

System.***out***.println("Returned Object reference ::"+System.*identityHashCode*( string.replaceAll("\\s+$", "") + "\""));

}// main

}// class

**Output ::**

Before replaceAll : " String with space "

String object reference before replaceAll ::918221580

Replace all space : "Stringwithspace"

Returned Object reference ::168423058

Replace all regex : "Stringwithspace"

Returned Object reference ::1450495309

Replace Leading : "String with space "

Returned Object reference ::1670782018

Replace trailing : " String with space"

Returned Object reference ::468121027

**Difference between replaceAll and replace method**

|  |  |
| --- | --- |
| **replaceAll()** | **replace()** |
| From Java 1.4 | From Java 1.5 |
| Accepts regular expression for target identification | Accepts string for target identification |
| Used for fix or dynamic string replacement | Used for fix string replacement |
| Removes characters having ASCII value less than or equal to ‘U+0020’ or ’32’ | Removes all space characters according to unicode |

**ReplaceFirst(String regex,String replacement) Method::**

* This method was introduced in jdk 1.4.
* This method can also can work with regular expression.
* Replaces the first substring of this string that matches the given [regular expression](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/regex/Pattern.html#sum) with the given replacement.

**Note** :: That backslashes (\) and dollar signs ($) in the replacement string may cause the results to be different than if it were being treated as a literal replacement string; see [Matcher.replaceAll](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/regex/Matcher.html" \l "replaceAll(java.lang.String)). Use [Matcher.quoteReplacement(java.lang.String)](https://docs.oracle.com/en/java/javase/14/docs/api/java.base/java/util/regex/Matcher.html" \l "quoteReplacement(java.lang.String)) to suppress the special meaning of these characters, if desired.

**Example 1 ::**

We just need to create correct regular expression with correct replacement parameter. Some regular expression examples as below:

|  |  |
| --- | --- |
| \s+ | Find all space |
| ^\s+ | Find all spaces at line beginning |
| \s+$ | Find all spaces at line ending |

**package** com.abhi.practice;

**public** **class** ReplaceFirstTest {

**public** **static** **void** main(String[] args) {

String string = " String with space ";

System.***out***.println("Before replaceFirst : \"" + string+"\"");

System.***out***.println("String object reference before replaceFirst ::"+System.*identityHashCode*(string));

System.***out***.println();

System.***out***.println("Replace First space : \"" + string.replaceFirst(" ", "") + "\"");

System.***out***.println("Returned Object reference ::"+System.*identityHashCode*(string.replaceFirst(" ", "") + "\""));

System.***out***.println();

System.***out***.println("Replace First regex : \"" + string.replaceFirst("\\s+", "") + "\"");

System.***out***.println("Returned Object reference ::"+System.*identityHashCode*(string.replaceFirst("\\s+", "") + "\""));

System.***out***.println();

System.***out***.println("Replace Leading : \"" + string.replaceFirst("^\\s+", "") + "\"");

System.***out***.println("Returned Object reference ::"+System.*identityHashCode*(string.replaceFirst("^\\s+", "") + "\""));

System.***out***.println();

System.***out***.println("Replace trailing : \"" + string.replaceFirst("\\s+$", "") + "\"");

System.***out***.println("Returned Object reference ::"+System.*identityHashCode*( string.replaceFirst("\\s+$", "") + "\""));

}// main

}

**Output ::**

Before replaceFirst : " String with space "

String object reference before replaceFirst ::918221580

Replace First space : " String with space "

Returned Object reference ::168423058

Replace First regex : "String with space "

Returned Object reference ::1450495309

Replace Leading : "String with space "

Returned Object reference ::1670782018

Replace trailing : " String with space"

Returned Object reference ::468121027