**Strings Java 11**

**New String APIs/Methods in Java 11 (JDK11) with Examples :**

Java 11 added a few useful APIs to the commonly used [String class](https://www.javaguides.net/2018/08/java-string-class-api-guide.html).

Here are the APIs/methods added to String class in Java 11:

***String java.lang.String.repeat(int count)*** - As the name suggests, the repeat() instance method repeats the string content.

***String java.lang.String.strip()*** - The strip() instance method returns a string with all leading and trailing whitespaces removed.

***String java.lang.String.stripLeading()*** - This method returns a string with all leading white space removed.

***String java.lang.String.stripTrailing()*** - This method returns a string with all trailing white space removed.

***booleanjava.lang.String.isBlank()*** - The isBlank() instance method returns true if the string is empty or contains only whitespace. Otherwise, it returns false.

***Stream java.lang.String.lines()*** - The lines() instance method returns a Stream of lines extracted from the string, separated by line terminators.

**repeat() API**

As the name suggests, the *repeat()* instance method repeats the string content.

It returns a string whose value is the concatenation of the string repeated *n* times, where *n* is passed as a parameter:

publicclassStringAPIInJDK11 {

publicstaticvoidmain(String[] args) {

String input ="Java";

/\*\*

\* the repeat() instance method repeats the string content

\*/

String output =input.repeat(2) +"Guides".repeat(3);

System.out.println(output); // Output: JavaJavaGuidesGuidesGuides

}

}

**Output:**

JavaJavaGuidesGuidesGuides

**strip(), stripLeading() and stripTrailing() methods**

**String java.lang.String.strip()** - The strip() instance method returns a string with all leading and trailing whitespaces removed.

**String java.lang.String.stripLeading()** - This method returns a string with all leading white space removed.

**String java.lang.String.stripTrailing()** - This method returns a string with all trailing white space removed.

publicclassStringAPIInJDK11 {

publicstaticvoidmain(String[] args) {

String input1 ="\n\t hello\u2005";

/\*\*

\* The strip() instance method returns a string

\* with all leading and trailing whitespaces removed

\*/

String output1 = input1.strip();

System.out.println(output1);

// stripLeading() usage

System.out.println(input1.stripLeading());

// stripTrailing() usage

System.out.println(input1.stripTrailing());

}

}

**Output:**

hello

hello ?

hello

**isBlank() API**

The *isBlank()* String class method returns true if the string is empty or contains only whitespace. Otherwise, it returns false:

publicclassStringAPIInJDK11 {

publicstaticvoidmain(String[] args) {

String input ="\n\t\u2005 ";

/\*\*

\* The isBlank() instance method returns true if the string is empty or contains

\* only whitespace. Otherwise, it returns false

\*/

boolean result =input.isBlank();

System.out.println(result);

}

}

**Output:**

true

**lines() API**

The *lines()* String class method returns a Stream of lines extracted from the string, separated by line terminators:

publicclassStringAPIInJDK11 {

publicstaticvoidmain(String[] args) {

/\*\*

\* The lines() instance method returns a Stream of lines extracted from the

\* string, separated by line terminators

\*/

String multilineStr="This is\n a multiline\n string.";

Stream<String> stream =multilineStr.lines();

stream.forEach(action ->System.out.println(action));

long lineCount=multilineStr.lines().count();

System.out.println(lineCount);

}

}

**Output:**

This is

a multiline

string.

3

A line terminator is one of the following: “\n”, “\r”, or “\r\n”.

The stream contains lines in the order in which they occur. The line terminator is removed from each line.