

commons-lang3

- description : commons-lang3
- author :
- email : dhan@repia.com
- lastupdate : 2020-03-24

MavanRepository

```
<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-lang3 -->
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-lang3</artifactId>
  <version>3.8</version>
</dependency>
```

MVN Repository

StringEscapeUtils

StringEscapeUtils (2020-04-06)

StringUtils

```
StringUtils.abbreviate ( , )
4
... ( 3 )
```

```
String abbreviate = StringUtils.abbreviate("ShowMeTheMoney", 7);
System.out.println(abbreviate);
// : Show...
```

```
StringUtils.appendIfMissing ( , 가 , )
가 , 가
```

```
String appendIfMissing = StringUtils.appendIfMissing("ShowMeTheMoney", "-
```

```

From.Choi", "Money");
System.out.println(appendIfMissing);
//           : ShowMeTheMoney

appendIfMissing = StringUtils.appendIfMissing("ShowMeTheMoney", "-
From.Choi", "test");
System.out.println(appendIfMissing);
//           : ShowMeTheMoney-From.Choi

```

StringUtils.appendIfMissingIgnoreCase (, 가 ,)

StringUtils.appendIfMissing

```

String appendIfMissingIgnoreCase =
StringUtils.appendIfMissingIgnoreCase("ShowMeTheMoney", "-From.Choi",
"money");
System.out.println(appendIfMissingIgnoreCase);
//           : ShowMeTheMoney

```

StringUtils.capitalize ()

```

String capitalize = StringUtils.capitalize("sss");
System.out.println(capitalize);
//           : Sss

```

StringUtils.chomp ()

\n, \r, \r\n

```

String chomp = StringUtils.chomp("sss\n");
System.out.println(chomp);
//           : SSS

```

StringUtils.chomp (,)

가

```

String chomp = StringUtils.chomp("sss#", "#");
System.out.println(chomp);
//           : SSS

```

StringUtils.chop ()

```
String chop = StringUtils.chop("sss");
System.out.println(chop);
//      : ss
```

StringUtils.center (,)

가 .
가 가 .

```
String center = StringUtils.center("sss", 5);
System.out.println("'" + center + "'");
//      : ' sss '
```

StringUtils.center (, , 가)

가 가 .
가 가 .

```
String center = StringUtils.center("sss", 6, "#");
System.out.println(center);
//      : #sss##
```

StringUtils.compare (,)

```
///가, ' '
System.out.println((int) '가');
System.out.println((int) ' ');
//      : 44032
//      : 45208

int compare = StringUtils.compare("가", " ");
System.out.println(compare);
//      : -1176
int compare2 = StringUtils.compare(" ", "가");
System.out.println(compare2);
//      : 1176
```

StringUtils.compareToIgnoreCase (,)

StringUtils.compare

```
int compareIgnoreCase = StringUtils.compareIgnoreCase("A", "a");
System.out.println(compareIgnoreCase);
//           : 0
```

```
StringUtils.contains (           ,           )
                               'true'
```

```
boolean contains = StringUtils.contains("Hello", "el");
System.out.println(contains);
//           : true
```

```
StringUtils.containsAny (           ,           ,           )
                               'true'
```

```
boolean containsAny = StringUtils.containsAny("           ", "           ");
System.out.println(containsAny);
//           : true
```

```
StringUtils.containsIgnoreCase (           ,           )
```

StringUtils.contains

```
boolean containsIgnoreCase = StringUtils.containsIgnoreCase("HELLO", "he");
System.out.println(containsIgnoreCase);
//           : true
```

```
StringUtils.containsNone (           ,           )
                               'false'
```

```
boolean containsNone = StringUtils.containsNone("hello", "aaah");
System.out.println(containsNone);
//           : false
```

```
StringUtils.containsOnly (           ,           )
```

'true'

```
boolean containsOnly = StringUtils.containsOnly("hello", "olleh");
System.out.println(containsOnly);
//      : true

boolean containsOnly2 = StringUtils.containsOnly("hello", "hell");
System.out.println(containsOnly2);
//      : false
```

StringUtils.containsWhitespace (,)

'true'

```
boolean containsWhitespace = StringUtils.containsWhitespace(" ");
System.out.println(containsWhitespace);
//      : true
```

StringUtils.countMatches (,)

1 , 0

```
int countMatches = StringUtils.countMatches("ABCD", "A");
System.out.println(countMatches);
//      : 1
int countMatches2 = StringUtils.countMatches("ABCD", "AB");
System.out.println(countMatches2);
//      : 1
int countMatches3 = StringUtils.countMatches("ABCD", "AC");
System.out.println(countMatches3);
//      : 0
```

StringUtils.defaultIfBlank (,)

space(), , null

```
String defaultIfBlank1 = StringUtils.defaultIfBlank(null, "A");
System.out.println(defaultIfBlank1);
//      : A
String defaultIfBlank2 = StringUtils.defaultIfBlank("", "A");
System.out.println(defaultIfBlank2);
//      : A
String defaultIfBlank3 = StringUtils.defaultIfBlank(" ", "A");
System.out.println(defaultIfBlank3);
//      : A
```


'true', 'false' .

```
boolean endsWith = StringUtils.endsWith("abc", ".");
System.out.println(endsWith);
//      : true
boolean endsWith = StringUtils.endsWith("abc", ".");
System.out.println(endsWith);
//      : false
```

StringUtils.endsWithAny (String, char, 1, char, 2, ...)

StringUtils.endsWith

'true' .

```
boolean endsWithAny = StringUtils.endsWithAny("AAA", "a");
System.out.println(endsWithAny);
//      : false
boolean endsWithAny = StringUtils.endsWithAny("AAA", "a", "A");
System.out.println(endsWithAny);
//      : true
```

StringUtils.endsWithIgnoreCase (String, String)

StringUtils.endsWith

```
boolean endsWithIgnoreCase = StringUtils.endsWithIgnoreCase("BBB", "b");
System.out.println(endsWithIgnoreCase);
//      : true
```

StringUtils.equals (String, String)

'true' .

```
boolean equals = StringUtils.equals("abcd", "abcd");
System.out.println(equals);
//      : true
boolean equals = StringUtils.equals("abcd", "a bcd");
System.out.println(equals);
//      : false
```

StringUtils.equalsAny (String, char, 1, char, 2, ...)

StringUtils.equals

.
'true'

```
boolean equalsAny = StringUtils.equalsAny("abcd", "a", "ab", "abcd");
System.out.println(equalsAny);
//           : true
```

StringUtils.equalsAnyIgnoreCase (, 1, 2, ...)

StringUtils.equalsAny

```
boolean equalsAnyIgnoreCase = StringUtils.equalsAnyIgnoreCase("abc",
"abcde", "ABC");
System.out.println(equalsAnyIgnoreCase);
//           : true
```

StringUtils.equalsIgnoreCase(,)

true
(/)

```
boolean equalsIgnoreCase = StringUtils.equalsIgnoreCase("ss", "sS");
System.out.println(equalsIgnoreCase);
//           : true
```

StringUtils.firstNonBlank(, ,)

null, , 가
StringUtils 3.8 가

```
String firstNonBlank = StringUtils.firstNonBlank(null, "", " ", "xyz");
System.out.println(firstNonBlank);
//           : xyz
```

StringUtils.getCommonPrefix(new String[] { " ", " " })

```
String getCommonPrefix = StringUtils.getCommonPrefix("sss", "ssS", "ssa");
System.out.println(getCommonPrefix);
//           : sS
```

```
StringUtils.getDigits("123456")
```

Unicode Number() .

```
String getDigits = StringUtils.getDigits("123456");
System.out.println(getDigits);
//      : 123456
```

```
StringUtils.indexOf("sS", "S")
```

```
int indexOf = StringUtils.indexOf("sS", "S");
System.out.println(indexOf);
//      : 1
```

```
StringUtils.indexOfAny("sS", new String[] {"s", "S"})
```

(vararg)

```
int indexOfAny = StringUtils.indexOfAny("sS", new String[] {"s", "S"});
System.out.println(indexOfAny);
//      : 0
```

```
StringUtils.indexOfAnyBut("sS", new char[] {'s', 'S'})
```

```
int indexOfAnyBut = StringUtils.indexOfAnyBut("sS", new char[] {'s', 'S'});
System.out.println(indexOfAnyBut);
//      : 2
```

```
StringUtils.indexOfDifference("sS", "S");
```

```
int indexOfDifference = StringUtils.indexOfDifference("sS", "S");
System.out.println(indexOfDifference);
```

```
// : 3
```

```
StringUtils.indexOfIgnoreCase(" ", " ")
```

```
( / )
```

```
int indexOfIgnoreCase = StringUtils.indexOfIgnoreCase("sS", "S");  
System.out.println(indexOfIgnoreCase);  
// : 0
```

Ref

- [StringUtils](#)
, [eleven](#), [stringutils](#), [commons-lang](#), [commons-lang3](#)

From: <http://rwiki.repia.com/> - . - 2023.12

Permanent link: <http://rwiki.repia.com/doku.php?id=wiki:java:commons-lang&rev=1586764102> 

Last update: **2022/03/10 19:52**