



链滴

StringUtils 常用方法

作者: [liuyishi](#)

原文链接: <https://ld246.com/article/1517559702348>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

public static void TestStr(){<br>
//null 和 ""操作~~~~~<br>
//判断是否 Null 或者 ""<br>
//System.out.println(StringUtils.isEmpty(null));<br>
//System.out.println(StringUtils.isNotEmpty(null));<br>
//判断是否 null 或者 "" 去空格~~~~~<br>
//System.out.println(StringUtils.isBlank(" ");<br>
//System.out.println(StringUtils.isNotBlank(null));<br>
//去空格.Null 返回 null~~~~~<br>
//System.out.println(StringUtils.trim(null));<br>
//去空格, 将 Null 和"" 转换为 Null<br>
//System.out.println(StringUtils.trimToNull(""));<br>
//去空格, 将 NULL 和 "" 转换为""<br>
//System.out.println(StringUtils.trimToEmpty(null));<br>
//可能是对特殊空格符号去除? ? <br>
//System.out.println(StringUtils.strip("大家好 啊 \t"));<br>
//同上, 将""和 null 转换为 Null<br>
//System.out.println(StringUtils.stripToNull("\t"));<br>
//同上, 将""和 null 转换为""<br>
//System.out.println(StringUtils.stripToEmpty(null));<br>
//将""或者 Null 转换为 ""<br>
//System.out.println(StringUtils.defaultString(null));<br>
//仅当字符串为 Null 时 转换为指定的字符串(二参数)<br>
//System.out.println(StringUtils.defaultString("", "df"));<br>
//当字符串为 null 或者""时, 转换为指定的字符串(二参数)<br>
//System.out.println(StringUtils.defaultIfEmpty(null, "sos"));<br>
//去空格.去字符~~~~~
~~~~~<br>
//如果第二个参数为 null 去空格(否则去掉字符串 2 边一样的字符, 到不一样为止)<br>
//System.out.println(StringUtils.strip("fsfsdf", "f"));<br>
//如果第二个参数为 null 只去前面空格(否则去掉字符串前面一样的字符, 到不一样为止)<br>
//System.out.println(StringUtils.stripStart("ddsuuu ", "d"));<br>
//如果第二个参数为 null 只去后面空格, (否则去掉字符串后面一样的字符, 到不一样为止)<br>
//System.out.println(StringUtils.stripEnd("dabads", "das"));<br>
//对数组没个字符串进行去空格。<br>
//ArrayToList(StringUtils.stripAll(new String[]{"中华", "民国", "共和"}));<br>
//如果第二个参数为 null.对数组每个字符串进行去空格。(否则去掉数组每个元素开始和结尾一样的符)<br>
//ArrayToList(StringUtils.stripAll(new String[]{"中华", "民国", "国共和国"}, "国"));<br>
//查找,判断~~~~~
~~~~~<br>
//判断 2 个字符串是否相等相等,Null 也相等<br>
//System.out.println(StringUtils.equals(null, null));<br>
//不区分大小写比较<br>
//System.out.println(StringUtils.equalsIgnoreCase("abc", "ABc"));<br>
//查找, 不知道怎么弄这么多查找, 很多不知道区别在哪? 费劲~~~~~
~~~~~<br>
//普通查找字符, 如果一参数为 null 或者""返回-1<br>
//System.out.println(StringUtils.indexOf(null, "a"));<br>
//从指定位置(三参数)开始查找, 本例从第 2 个字符开始查找 k 字符<br>
//System.out.println(StringUtils.indexOf("akfekcd 中华", "k", 2));<br>
//未发现不同之处<br>
//System.out.println(StringUtils.ordinalIndexOf("akfekcd 中华", "k", 2));<br>
//查找,不区分大小写<br>

```

```

//System.out.println(StringUtils.indexOfIgnoreCase("adfs", "D"));<br>
//从指定位置(三参数)开始查找,不区分大小写<br>
//System.out.println(StringUtils.indexOfIgnoreCase("adfs", "a", 3));<br>
//从后往前查找<br>
//System.out.println(StringUtils.lastIndexOf("adfas", "a"));<br>
//未理解,此结果为 2<br>
//System.out.println(StringUtils.lastIndexOf("d 饿 abasdafs 我", "a", 3));<br>
//未解,此结果为-1<br>
//System.out.println(StringUtils.lastOrdinalIndexOf("yksdfdht", "f", 2));<br>
//从后往前查, 不区分大小写<br>
//System.out.println(StringUtils.lastIndexOfIgnoreCase("sdffet", "E"));<br>
//未解,此结果为 1<br>
//System.out.println(StringUtils.lastIndexOfIgnoreCase("efefrfs 看", "F", 2));<br>
//检查是否查到, 返回 boolean,null 返回假<br>
//System.out.println(StringUtils.contains("sdf", "dg"));<br>
//检查是否查到, 返回 boolean,null 返回假,不区分大小写<br>
//System.out.println(StringUtils.containsIgnoreCase("sdf", "D"));<br>
//检查是否有含有空格,返回 boolean<br>
//System.out.println(StringUtils.containsWhitespace(" d"));<br>
//查询字符串跟数组任一元素相同的第一次相同的位置<br>
//System.out.println(StringUtils.indexOfAny("absfekf", new String[]{"f", "b"}));<br>
//查询字符串中指定字符串(参数二)出现的次数<br>
//System.out.println(StringUtils.indexOfAny("afefes", "e"));<br>
//查找字符串中是否有字符数组中相同的字符, 返回 boolean<br>
//System.out.println(StringUtils.containsAny("asfsd", new char[]{'k', 'e', 's'}));<br>
//未理解与 lastIndexOf 不同之处。是否查到, 返回 boolean<br>
//System.out.println(StringUtils.containsAny("啡 f 咖啡", "咖"));<br>
//未解<br>
//System.out.println(StringUtils.indexOfAnyBut("seefaff", "af"));<br>
//判断字符串中所有字符, 都是出自参数二中。<br>
//System.out.println(StringUtils.containsOnly("中华华", "华"));<br>
//判断字符串中所有字符, 都是出自参数二的数组中。<br>
//System.out.println(StringUtils.containsOnly("中华中", new char[]{'中', '华'}));<br>
//判断字符串中所有字符, 都不在参数二中。<br>
//System.out.println(StringUtils.containsNone("中华华", "国"));<br>
//判断字符串中所有字符, 都不在参数二的数组中。<br>
//System.out.println(StringUtils.containsNone("中华中", new char[]{'中', '达人'}));<br>
//从后往前查找字符串中与字符数组中相同的元素第一次出现的位置。本例为 4<br>
//System.out.println(StringUtils.lastIndexOfAny("中华人民共和国", new String[]{"国人", "共和"}));<br>
//未发现与 indexOfAny 不同之处 查询字符串中指定字符串(参数二)出现的次数<br>
//System.out.println(StringUtils.countMatches("中国人民共和中国", "中国"));<br>
//检查是否 CharSequence 的只包含 Unicode 的字母。空将返回 false。一个空的 CharSequence 长 () = 0) 将返回 true<br>
//System.out.println(StringUtils.isAlpha("这是干什么的 2"));<br>
//检查是否只包含 Unicode 的 CharSequence 的字母和空格 (")。空将返回一个空的 CharSequence 假 (长 () = 0) 将返回 true。<br>
//System.out.println(StringUtils.isAlphaSpace("NBA 直播"));<br>
//检查是否只包含 Unicode 的 CharSequence 的字母或数字。空将返回 false。一个空的 CharSequence (长 () = 0) 将返回 true。<br>
//System.out.println(StringUtils.isAlphanumeric("NBA 直播"));<br>
//如果检查的 Unicode CharSequence 的只包含字母, 数字或空格 (")。空将返回 false。一个空的 CharSequence (长 () = 0) 将返回 true。<br>
//System.out.println(StringUtils.isAlphanumericSpace("NBA 直播"));<br>

```

```

//检查是否只包含 ASCII 可 CharSequence 的字符。空将返回 false。一个空的 CharSequence (长
) = 0) 将返回 true。 <br>
//System.out.println(StringUtils.isAsciiPrintable("NBA 直播")); <br>
//检查是否只包含数值。 <br>
//System.out.println(StringUtils.isNumeric("NBA 直播")); <br>
//检查是否只包含数值或者空格 <br>
//System.out.println(StringUtils.isNumericSpace("33 545")); <br>
//检查是否只是空格或"。 <br>
//System.out.println(StringUtils.isWhitespace(" ")); <br>
//检查是否全是英文小写。 <br>
//System.out.println(StringUtils.isAllLowerCase("kjk33")); <br>
//检查是否全是英文大写。 <br>
//System.out.println(StringUtils.isAllUpperCase("KJKJ")); <br>
//交集操作~~~~~
~~~~~ <br>
//去掉参数 2 字符串中在参数一中开头部分共有的部分， 结果为:人民共和加油 <br>
//System.out.println(StringUtils.difference("中国加油", "中国人民共和加油")); <br>
//统计 2 个字符串开始部分共有的字符个数 <br>
//System.out.println(StringUtils.indexOfDifference("www.taobao", "www.taobao.com")); <br>
//统计数组中各个元素的字符串开始都一样的字符个数 <br>
//System.out.println(StringUtils.indexOfDifference(new String[] {"中国加油", "中国共和", "中国
民"})); <br>
//取数组每个元素共同的部分字符串 <br>
//System.out.println(StringUtils.getCommonPrefix(new String[] {"中国加油", "中国共和", "中国
民"})); <br>
//统计参数一中每个字符与参数二中每个字符不同部分的字符个数 <br>
//System.out.println(StringUtils.getLevenshteinDistance("中国共和发国人民", "共和国")); <br>
//判断开始部分是否与二参数相同 <br>
//System.out.println(StringUtils.startsWith("中国共和国人民", "中国")); <br>
//判断开始部分是否与二参数相同。不区分大小写 <br>
//System.out.println(StringUtils.startsWithIgnoreCase("中国共和国人民", "中国")); <br>
//判断字符串开始部分是否与数组中的某一元素相同 <br>
//System.out.println(StringUtils.startsWithAny("abef", new String[]{"ge", "af", "ab"})); <br>
//判断结尾是否相同 <br>
//System.out.println(StringUtils.endsWith("abcdef", "def")); <br>
//判断结尾是否相同，不区分大小写 <br>
//System.out.println(StringUtils.endsWithIgnoreCase("abcdef", "Def")); <br>
//字符串截取~~~~~
~ <br>
//截取指定位置的字符， null 返回 null."返回" <br>
//System.out.println(StringUtils.substring("国民党", 2)); <br>
//截取指定区间的字符 <br>
//System.out.println(StringUtils.substring("中华人民共和国", 2, 4)); <br>
//从左截取指定长度的字符串 <br>
//System.out.println(StringUtils.left("说点什么好呢", 3)); <br>
//从右截取指定长度的字符串 <br>
//System.out.println(StringUtils.right("说点什么好呢", 3)); <br>
//从第几个开始截取，三参数表示截取的长度 <br>
//System.out.println(StringUtils.mid("说点什么好呢", 3, 2)); <br>
//截取到等于第二个参数的字符串为止 <br>
//System.out.println(StringUtils.substringBefore("说点什么好呢", "好")); <br>
//从左往右查到相等的字符开始，保留后边的，不包含等于的字符。本例：什么好呢 <br>
//System.out.println(StringUtils.substringAfter("说点什么好呢", "点")); <br>
//这个也是截取到相等的字符，但是是从右往左。本例结果：说点什么好 <br>

```

```

//System.out.println(StringUtils.substringBeforeLast("说点什么好点呢", "点"));<br>
//这个截取同上是从右往左。但是保留右边的字符<br>
//System.out.println(StringUtils.substringAfterLast("说点什么好点呢?", "点"));<br>
//截取查找到第一次的位置, 和第二次的位置中间的字符。如果没找到第二个返回 null。本例结果:20
0 世界杯在<br>
//System.out.println(StringUtils.substringBetween("南非 2010 世界杯在南非, 在南非", "南非"));
br>
//返回参数二和参数三中间的字符串, 返回数组形式<br>
//ArrayToList(StringUtils.substringsBetween("[a][b][c]", "[", "]"));<br>
//分割~~~~~
~<br>
//用空格分割成数组, null 为 null<br>
//ArrayToList(StringUtils.split("中华 人民 共和"));<br>
//以指定字符分割成数组<br>
//ArrayToList(StringUtils.split("中华,人民,共和", ","));<br>
//以指定字符分割成数组, 第三个参数表示分隔成数组的长度, 如果为 0 全体分割<br>
//ArrayToList(StringUtils.split("中华: 人民: 共和", ":", 2));<br>
//未发现不同的地方,指定字符分割成数组<br>
//ArrayToList(StringUtils.splitByWholeSeparator("ab-!-cd-!-ef", "-!-"));<br>
//未发现不同的地方,以指定字符分割成数组, 第三个参数表示分隔成数组的长度<br>
//ArrayToList(StringUtils.splitByWholeSeparator("ab-!-cd-!-ef", "-!-", 2));<br>
//分割, 但" "不会被忽略算一个元素,二参数为 null 默认为空格分隔<br>
//ArrayToList(StringUtils.splitByWholeSeparatorPreserveAllTokens(" ab de fg ", null));<br>
//同上, 分割, " "不会被忽略算一个元素。第三个参数代表分割的数组长度。<br>
//ArrayToList(StringUtils.splitByWholeSeparatorPreserveAllTokens("ab de fg", null, 3));<br>
//未发现不同地方,分割<br>
//ArrayToList(StringUtils.splitPreserveAllTokens(" ab de fg "));<br>
//未发现不同地方,指定字符分割成数组<br>
//ArrayToList(StringUtils.splitPreserveAllTokens(" ab de fg ", null));<br>
//未发现不同地方,以指定字符分割成数组, 第三个参数表示分隔成数组的长度<br>
//ArrayToList(StringUtils.splitPreserveAllTokens(" ab de fg ", null, 2));<br>
//以不同类型进行分隔<br>
//ArrayToList(StringUtils.splitByCharacterType("AEkjKr i39:。中文"));<br>
//未解<br>
//ArrayToList(StringUtils.splitByCharacterTypeCamelCase("ASFsRules234"));<br>
//拼接~~~~~<br>
//将数组转换为字符串形式<br>
//System.out.println(StringUtils.concat(getArrayData()));<br>
//拼接时用参数一得字符相连接.注意 null 也用连接符连接了<br>
//System.out.println(StringUtils.concatWith(",", getArrayData()));<br>
//也是拼接。未发现区别<br>
//System.out.println(StringUtils.join(getArrayData()));<br>
//用连接符拼接, 为发现区别<br>
//System.out.println(StringUtils.join(getArrayData(), "。"));<br>
//拼接指定数组下标的开始(三参数)和结束(四参数,不包含)的中间这些元素, 用连接符连接<br>
//System.out.println(StringUtils.join(getArrayData(), ":", 1, 3));<br>
//用于集合连接字符串.用于集合<br>
//System.out.println(StringUtils.join(getListData(), "。"));<br>
//移除, 删除~~~~~
~~~~~<br>
//删除所有空格符<br>
//System.out.println(StringUtils.deleteWhitespace(" s 中 你 4j"));<br>
//移除开始部分的相同的字符<br>
//System.out.println(StringUtils.removeStart("www.baidu.com", "www."));<br>

```



```

//移除开始部分的相同的字符,不区分大小写 <br>
//System.out.println(StringUtils.removeStartIgnoreCase("www.baidu.com", "WWW"));<br>
//移除后面相同的部分 <br>
//System.out.println(StringUtils.removeEnd("www.baidu.com", ".com"));<br>
//移除后面相同的部分, 不区分大小写 <br>
//System.out.println(StringUtils.removeEndIgnoreCase("www.baidu.com", ".COM"));<br>
//移除所有相同的部分 <br>
//System.out.println(StringUtils.remove("<a href='https://ld246.com/forward?goto=http%3
%2F%2Fwww.baidu.com%2Fbaidu%2522' target='_blank' rel='nofollow ugc'>www.baidu.c
m/baidu" </a>", "bai"));<br>
//移除结尾字符为"\n", "\r", 或者 "\r\n".<br>
//System.out.println(StringUtils.chomp("abcabc\r"));<br>
//也是移除, 未解。去结尾相同字符 <br>
//System.out.println(StringUtils.chomp("baidu.com", "com"));<br>
//去掉末尾最后一个字符.如果是"\n", "\r", 或者 "\r\n"也去除 <br>
//System.out.println(StringUtils.chop("www.baidu"));<br>
//替换~~~~~<br>
~~~~~<br>
//替换指定的字符, 只替换第一次出现的<br>
//System.out.println(StringUtils.replaceOnce("<a href='https://ld246.com/forward?goto=htt
%3A%2F%2Fwww.baidu.com%2Fbaidu%2522' target='_blank' rel='nofollow ugc'>www.bai
u.com/baidu" </a>", "baidu", "hao123"));<br>
//替换所有出现过的字符 <br>
//System.out.println(StringUtils.replace("<a href='https://ld246.com/forward?goto=http%3A
2F%2Fwww.baidu.com%2Fbaidu%2522' target='_blank' rel='nofollow ugc'>www.baidu.com
baidu" </a>", "baidu", "hao123"));<br>
//也是替换, 最后一个参数表示替换几个 <br>
//System.out.println(StringUtils.replace("<a href='https://ld246.com/forward?goto=http%3A
2F%2Fwww.baidu.com%2Fbaidu%2522' target='_blank' rel='nofollow ugc'>www.baidu.com
baidu" </a>", "baidu", "hao123", 1));<br>
//这个有意识, 二三参数对应的数组, 查找二参数数组一样的值, 替换三参数对应数组的值。本例:bai
u 替换为 taobao。com 替换为 net<br>
//System.out.println(StringUtils.replaceEach("<a href='https://ld246.com/forward?goto=http
3A%2F%2Fwww.baidu.com%2Fbaidu%2522' target='_blank' rel='nofollow ugc'>www.baidu.
om/baidu" </a>", new String[]{"baidu", "com"}, new String[]{"taobao", "net"}));<br>
//同上, 未发现不同 <br>
//System.out.println(StringUtils.replaceEachRepeatedly("<a href='https://ld246.com/forward
goto=http%3A%2F%2Fwww.baidu.com%2Fbaidu%2522' target='_blank' rel='nofollow ugc'
www.baidu.com/baidu" </a>", new String[]{"baidu", "com"}, new String[]{"taobao", "net"}));<br>

//这个更好, 不是数组对应, 是字符串参数二和参数三对应替换.(二三参数不对应的话, 自己看后果)<
r>
//System.out.println(StringUtils.replaceChars("www.baidu.com", "bdm", "qo"));<br>
//替换指定开始(参数三)和结束(参数四)中间的所有字符 <br>
//System.out.println(StringUtils.overlay("www.baidu.com", "hao123", 4, 9));<br>
//添加, 增加~~~~~<br>
~~~~~<br>
//复制参数一的字符串, 参数二为复制的次数 <br>
//System.out.println(StringUtils.repeat("ba", 3));<br>
//复制参数一的字符串, 参数三为复制的次数。参数二为复制字符串中间的连接字符串 <br>
//System.out.println(StringUtils.repeat("ab", "ou", 3));<br>
//如何字符串长度小于参数二的值, 末尾加空格补全。(小于字符串长度不处理返回) <br>
//System.out.println(StringUtils.rightPad("海川", 4));<br>
//字符串长度小于二参数, 末尾用参数三补上, 多余的截取(截取补上的字符串) <br>

```

```
//System.out.println(StringUtils.rightPad("海川", 4, "河流啊"));<br>
//同上在前面补全空格<br>
//System.out.println(StringUtils.leftPad("海川", 4));<br>
//字符串长度小于二参数，前面用参数三补上，多于的截取(截取补上的字符串)<br>
//System.out.println(StringUtils.leftPad("海川", 4, "大家好"));<br>
//字符串长度小于二参数。在两侧用空格平均补全（测试后面补空格优先）<br>
//System.out.println(StringUtils.center("海川", 3));<br>
//字符串长度小于二参数。在两侧用三参数的字符串平均补全（测试后面补空格优先）<br>
//System.out.println(StringUtils.center("海川", 5, "流"));<br>
//只显示指定数量(二参数)的字符,后面以三个点补充(参数一截取 + 三个点=二参数)<br>
//System.out.println(StringUtils.abbreviate("中华人民共和国", 5));<br>
//2 头加点这个有点乱。本例结果: ...ijklmno<br>
//System.out.println(StringUtils.abbreviate("abcdefghijklmno", 12, 10));<br>
//保留指定长度，最后一个字符前加点.本例结果: ab.f<br>
//System.out.println(StringUtils.abbreviateMiddle("abcdef", ".", 4));<br>
//转换,刷选~~~~~<br>
~~~~~<br>
//转换第一个字符为大写.如何第一个字符是大写原始返回<br>
//System.out.println(StringUtils.capitalize("Ddf"));<br>
//转换第一个字符为大写.如何第一个字符是大写原始返回<br>
//System.out.println(StringUtils.uncapitalize("DTf"));<br>
//反向转换，大写变小写，小写变大写<br>
//System.out.println(StringUtils.swapCase("I am Jiang, Hello"));<br>
//将字符串倒序排列<br>
//System.out.println(StringUtils.reverse("中国人民"));<br>
//根据特定字符(二参数)分隔进行反转<br>
//System.out.println(StringUtils.reverseDelimited("中:国:人民", ':'));<br>
}</p>
```