

# 字符串数组的配套函数

## 讲义和代码可以看配套的课程的第一个视频下载

配套课程b站：《MATLAB教程新手入门篇（数学建模清风主讲，适合零基础同学观看）》

数学建模清风老师版权所有

## 背景

自MATLAB 2016b版本正式引入字符串类型起，其文本处理能力就得到了进一步提升。为了增强字符串数组的操控性和便利性，MATLAB精心提供了一系列与之配套的文本处理函数。

在本章前面的小节中，我们已经介绍了其中的一些函数。

函数名称	函数功能	章节位置
<b>newLine</b>	创建换行符，等价于 char(10)	5.2.2 节
<b>strip</b>	删除文本中的前导和尾部字符	5.2.3 节
<b>replace</b>	将旧文本替换成新的文本，不允许重叠子文本	5.3.2 节
<b>strings</b>	创建空的字符串数组	5.4.1 节
<b>string</b>	将其他数据类型转换为字符串数组	5.4.1 节
<b>strlength</b>	计算字符串中包含的字符数量	5.4.1 节
<b>reverse</b>	反转字符串中的字符顺序	5.4.2 节
<b>isstring</b>	确定输入数据是否为字符串数组类型	本章附录 3

## 要介绍的新函数

这些函数根据功能可划分为六大板块，我们先给出相应的表格

函数名称	函数功能	所属板块
<b>pad</b>	为字符串添加前导或尾随字符，以达到特定长度	调整功能
<b>join</b>	连接字符串数组中的连续元素，可指定维度	连接和拆分功能
<b>split</b>	根据指定分隔符拆分字符串	
<b>splitlines</b>	根据换行符拆分字符串	
<b>replaceBetween</b>	在指定的起点和终点之间替换字符串	替换和插入功能
<b>insertAfter</b>	在指定的子字符串后插入新的字符串	
<b>insertBefore</b>	在指定的子字符串前插入新的字符串	
<b>extractBetween</b>	提取起点和终点之间的子字符串	提取功能
<b>extractAfter</b>	提取指定位置后的子字符串	
<b>extractBefore</b>	提取指定位置前的子字符串	
<b>erase</b>	删除字符串内的特定子字符串	删除功能
<b>eraseBetween</b>	删除起点和终点之间的子字符串	
<b>contains</b>	判断字符串中是否包含特定模式	查找功能
<b>count</b>	统计字符串中特定模式出现的次数	
<b>startsWith</b>	判断字符串是否以特定的模式开头	
<b>endsWith</b>	判断字符串是否以特定的模式结尾	

## 调整功能

<pre>s = ["abcdefg";     "hij ";     "kl "];</pre>	3×1 string 数组 "abcdefg" "hij " "kl "
<pre>strjust(s) % 默认右对齐 % strjust(s,'right')</pre>	3×1 string 数组 "abcdefg" "hij " " kl"
<pre>strjust(s,'center') % 居中对齐</pre>	3×1 string 数组 "abcdefg" " hij " " kl "
<pre>strjust(s,'left') % 左对齐</pre>	3×1 string 数组 "abcdefg" "hij " "kl "

<pre>s = ["abcdefg";     "hij ";     "kl "]; pad(s)</pre>	3×1 string 数组 "abcdefg" "hij " "kl "
---	---

<pre>s = ["33";     "666";     "8888"]; pad(s,10)</pre>	3×1 string 数组 "33 " "666 " "8888 "
---	---

<pre>s = ["1";     "1111";     "11111";     "1111111"]; strjust(s,'center') % 没有空格不会调整</pre>	4×1 string 数组 "1" "111" "11111" "1111111"
--	---

<pre>pad(s,'both') % 不是 center 哦</pre>	4×1 string 数组 " 1 " " 111 " " 1111 " "1111111"
--	--

<pre>% s 同上 pad(s,15,'both')</pre>	4×1 string 数组 " 1 " " 111 " " 11111 " "1111111"
------------------------------------	---

% 不能颠倒输入参数的顺序  
pad(s,'both',15)

报错: Padding 必须为文本标量。

<pre>% s 同上 pad(s,'*')</pre>	4×1 string 数组 "1*****" "111*****" "11111*****" "1111111*****"
------------------------------	---

<pre>pad(s,'both','*')</pre>	4×1 string 数组 "*****1*****" "*****111*****" "*****1111*****" "*****11111*****"
------------------------------	--

% 只能将 padCharacter 参数作为最后一个输入  
pad(s,'\*','both')

报错: 'side' 选项必须为 'left', 'right', 或 'both'。

## 连接和拆分功能

join函数是一个强大的工具，它可以将字符串数组中的元素按指定的维度进行连接。

<pre>s = ["ab","cd";     "Ab","ce";     "c","cd"]; strjoin(s)</pre>	"ab Ab c cd ce cd"
<pre>strjoin(s,' ')</pre>	"ab, Ab, c, cd, ce, cd"
<pre>join(s) % 沿列方向添加空格进行连接 % 等价于 join(s,2) 即 dim = 2</pre>	3×1 string 数组 "ab cd" "Ab ce" "c cd"
<pre>join(s,1) % 沿行方向添加空格进行连接 % 这里的 1 表示 dim = 1</pre>	1×2 string 数组 "ab Ab c" "cd ce cd"
<pre>join(s',' ',1) % 指定用于连接的分隔符 % 指定的分隔符需要放在 dim 的前面</pre>	1×2 string 数组 "ab, Ab, c" "cd, ce, cd"
<pre>join(s',' ',2) % 通常简写为 join(s,' ')</pre>	3×1 string 数组 "ab, cd" "Ab, ce" "c, cd"

如果要连接的文本是1×N或N×1大小的字符串向量，则 strjoin函数和join函数的效果相同

如果要连接的数组是一维的字符串向量元胞数组，使用strjoin函数更为方便，因为它返回一个字符串向量，而join函数会返回一个包含单个元素的字符串向量元胞数组。

在5.3.2小节中，我们已经介绍了strsplit函数，这个函数只能用于处理字符串向量或字符串标量。然而，split函数提供了更广泛的应用范围，支持字符串数组、字符串向量，以及字符串向量元胞数组的处理。这使得split函数在处理多维文本数据时更加灵活和强大。

<pre>s = ["Mary Butler";     "Santiago Marquez";     "Diana Lee"]; split(s)</pre>	3×2 string 数组 "Mary" "Butler" "Santiago" "Marquez" "Diana" "Lee"
---	---

strsplit(s)

报错: 第一个输入必须为字符串向量或字符串标量。

s中的每个元素必须可拆分成相同数量的子字符串，否则MATLAB会报错。

如果数量不同，我们可以使用for循环进行处理，并将拆分结果保存到元胞数组中。

(3) splitlines函数  
splitlines函数用于在换行符处拆分字符串，即它固定了分隔符delimiter为换行符，因此我们可以使用splitlines(s)来代替split(s,newline)

## 替换和插入功能

以str等于"ABCDEFGHJI"为例，假设我们想将子字符串"DEF"替换成"\*\*\*"，我们可以通过以下两种方式指定其位置：

replaceBetween(str,"C","G","*")	即C和G之间
replaceBetween(str,4,6,"*")	即第4至第6个位置

<pre>str = "ABCDEFGHJI"; replaceBetween(str,"C","G","*") % 你也可以写成: % replaceBetween(str,"BC","G","*") % 不会替换 startPat 和 endPat 本身 replaceBetween(str,4,6,"*") % startPos 和 endPos 这两个位置上的字符也会被替换</pre>	"ABC*GHIJ"
	"ABC*GHIJ"

再来看一个例子，s 中保存着一些单词，假如我们想将 s 中每个单词最后的两个字符替换成"\*\*\*"，我们可以这样做：

<pre>s = ["great", "good", "perfect"]; n = strlength(s); replaceBetween(s,n-1,n,"**")</pre>	1×3 string 数组 "gre***" "go***" "perfe***"
---	--

上方代码中，变量n表示字符串数组s中各元素的长度。由于我们的目标是替换每个字符串的最后两个字符，因此起始位置被设定为n-1（每个字符串长度减1，即倒数第二个字符的位置），结束位置为n（即每个字符串的最后一个字符的位置）。

我们可通过'Boundaries'参数强制包含或不包含指定的开始和结束位置。当 bounds 为'inclusive'时，包含开始和结束位置；当 bounds 为'exclusive'时，不包含开始和结束位置。

<pre>str = "ABCDEFGHJIABCDGHIJ"; replaceBetween(str,"C","G","*","...","Boundaries','inclusive') % 默认不包含，使用'Boundaries'参数设置为包含 str = "ABCDEFGHJIABCDGHIJ"; replaceBetween(str,4,6,"*","...","Boundaries','exclusive') % 默认包含，使用'Boundaries'参数设置为不包含</pre>	"AB*HIIAB*HIJ"
	"ABCD*FGHIJABCDGHIJ"

insertAfter函数可以在指定的子字符串后插入新的字符串，和replaceBetween函数类似，子字符串的位置可以通过两种方式指定

insertBefore函数和insertAfter函数的用法完全相同，它可以在指定的子字符串前插入字符串。

<pre>s = ["香蕉 (3)";     "苹果 (6)";     "菠萝蜜 (10)"]; s1 = insertAfter(s, " (" ,"价格: ") % 在左括号 (后面插入新的字符串</pre>	3×1 string 数组 "香蕉 (价格: 3) " "苹果 (价格: 6) " "菠萝蜜 (价格: 10) "
<pre>s2 = insertBefore(s1,") " ,"元/斤") % 在右括号 ) 前面插入新的字符串</pre>	3×1 string 数组 "香蕉 (价格: 3 元/斤) " "苹果 (价格: 6 元/斤) " "菠萝蜜 (价格: 10 元/斤) "
<pre>s = "abc def ghi"; insertAfter(s, " " "   ") s = ["abcd","efgh"]; insertAfter(s,2," ") % 等价于 insertBefore(s,3," ")</pre>	"abc  def   ghi"
<pre>s = ["aaa","a12345"]; insertAfter(s,strlength(s),".txt") % 等价于直接计算 s + ".txt"</pre>	1×2 string 数组 "aaa.txt" "a12345.txt"

## 提取功能

<pre>str = "ABCDEFGHJIAB12345IJ"; extractBetween(str,"AB","I")</pre>	2×1 string 数组 "CDEFGH" "12345"
<pre>extractBetween(str,"AB","I", ...     'Boundaries','inclusive')</pre>	2×1 string 数组 "ABCDEFGHI" "AB12345I"
<pre>extractBetween(str,3,5)</pre>	"CDE"
<pre>str = ["Edgar Allen Poe";     "Louisa May Alcott"]; extractBetween(str,[7;8],[11;10])</pre>	2×1 string 数组 "Allen" "May"

这两个函数分别用于提取指定位置后和指定位置前的子字符串，它们的使用方法类似于insertAfter函数和insertBefore函数。

<pre>str = "ABCDEFGHJIAB12345IJ"; extractAfter(str,"A")</pre>	"BCDEFGHJIAB12345IJ"
<pre>extractBefore(str,"A")</pre>	""
<pre>extractBefore(str,3)</pre>	"AB"
<pre>extractAfter(str,3)</pre>	"DEFGHJIAB12345IJ"
<pre>str = ["abcde","fghijklmn"]; % 提取最后两个字符 extractAfter(str,strlength(str)-2)</pre>	1×2 string 数组 "de" "mn"

## 删除功能

erase函数用于删除字符串内的子字符串，它的用法如下：

newStr = erase(str,match) 删除 str 中 match 的所有实例。erase 函数将其余文本返回为 newStr。

<pre>str = "good mood omg"; erase(str,"o")</pre>	"gd md mg"
<pre>erase(str,["o", " "])</pre>	"gdmimg"
<pre>str = ["射雕英雄传.txt";     "鹿鼎记.docx";     "倚天屠龙记.doc";     "天龙八部.txt"]; erase(str,[".",".txt",".doc",".docx"])</pre>	4×1 string 数组 "射雕英雄传" "鹿鼎记" "倚天屠龙记" "天龙八部"
<pre>erase(str,[".",".txt",".docx",".doc"])</pre>	4×1 string 数组 "射雕英雄传" "鹿鼎记" "倚天屠龙记" "天龙八部"
<pre>music = ["光辉岁月 (来自某某公众号).mp3";     "海阔天空 (来自某某公众号).mp3";     "喜欢你 (来自某某公众号).mp3"]; % 去除文件名中的广告 erase(music, " (来自某某公众号) ")</pre>	3×1 string 数组 "光辉岁月.mp3" "海阔天空.mp3" "喜欢你.mp3"

eraseBetween函数用于删除起点和终点之间的子字符串，它的用法如下：

newStr = eraseBetween(str,startPat,endPat) 删除 str 中出现在子字符串 startPat 和 endPat 之间的所有字符

<pre>str = "abcdefg ab12345fghi"; eraseBetween(str,"b","f")</pre>	"abfg abfghi"
<pre>eraseBetween(str,"b","h")</pre>	"abhi"
<pre>eraseBetween(str,3,5)</pre>	"abfg ab12345fghi"
<pre>str = "&lt;html&gt;&lt;body&gt;&lt;h1&gt;Hello," + ...     " World!&lt;/h1&gt;&lt;/body&gt;&lt;/html&gt;"; % 删除 HTML 标签 eraseBetween(str,"&lt;","&gt;") eraseBetween(str,"&lt;","&gt;","...","Boundaries','inclusive')</pre>	"<<<<Hello, World!<<<<"
	"Hello, World!"

contains函数用于确定字符串中是否有模式，它的用法如下：

如果 str 包含指定的模式，TF = contains(str,pat) 将返回 1 (true)，否则返回 0 (false)。

<pre>str = ["abcDe","aaccef","ad"]; contains(str,"d")</pre>	1×3 logical 数组 0 0 1
<pre>contains(str,["f","d"], ...     'IgnoreCase',true)</pre>	1×3 logical 数组 1 1 1
<pre>files = ["image.png", "bbb.txt", ...     "data.csv", "bbb.txt"]; ind = contains(files, ".txt") % 返回所有以 ".txt" 结尾的文件名 txtFiles = files(ind)</pre>	1×4 logical 数组 0 1 0 1
	1×2 string 数组 "aaa.txt" "bbb.txt"

count函数用于计算字符串中模式的出现次数，它的用法如下：

A = count(str,pat) 返回 pat 在 str 中的出现次数。

<pre>str = ["red green blue";     "green red blue green blue"]; count(str,"blue")</pre>	1 2
<pre>count(str,["red","blue"])</pre>	2 3
<pre>str = ["abababab"]; count(str,"aba") % 不允许重叠子文本</pre>	2
<pre>s = ["我永远 18 岁";     "今天天气很好！";     "你好,886!";     "Hi,520-1314"]; ss = string(0:9); count(ss)</pre>	2 0 3 7

startsWith函数和endsWith函数分别用于判断字符串是否以特定的模式开头或结尾，以startsWith函数为例，它的使用方法如下：

如果 str 以指定的模式开头，则 TF = startsWith(str,pat) 返回 1 (true)，否则返回 0 (false)。

<pre>s = ["刘诗诗","杨幂","刘亦菲"]; startsWith(s,"刘")</pre>	1×3 logical 数组 1 0 1
<pre>str = ["aa.png","bb.jpg","cc.txt"]; endsWith(str,[".jpg",".png"])</pre>	1×3 logical 数组 1 1 0