

# VIM学习笔记 环绕字符编辑(surround)

**surround** 插件可以快速编辑围绕在内容两端的字符（pairs of things surrounding things），比如成对出现的括号、引号，甚至HTML/XML标签等。

## 安装配置

推荐您使用[vim-plug](#)或[Vundle](#)等插件管理器，来安装GitHub上的[surround](#)插件。

以下将利用实例来介绍surround插件的主要功能，请注意：

- 在“原始文本”列中，高亮文字表示光标所在位置；
- 在“命令”列中，为顺序执行的命令序列。
- 在“更改效果”列中，为命令执行之后的结果。请参考[实例文件](#)并自行测试。

## 新增环绕字符

在常规模式、插入模式和可视化模式下，可以分别使用**ys**和**S**命令来新增环绕字符：

模式	目标范围	新增内容	原始文本	命令	更改效果
常规模式	单词	字符	Hello World	<b>ysi*</b>	Hello *World*
	单词	标签	Hello World	<b>ysiwt&lt;strong&gt;</b>	Hello <strong>World</strong>
	单词	括号	print var1, var2	<b>ys3w)</b>	print (var1, var2)
	行尾	括号	print var1, var2	<b>ys\$)</b>	print (var1, var2)
	查找	标签	William Shakespeare said, "Brevity is the soul of wit."	<b>ysf"t&lt;em&gt;</b>	William Shakespeare said, <em>"Brevity is the soul of wit."</em>
	整行	字符	Hello World	<b>yss"</b>	"Hello World"
	整行	标签	Hello World	<b>ysst&lt;p&gt;</b>	<p>Hello World</p>

	整行	新行	Hello World	ySS"	" Hello World "
插入模式		引号		<CTRL-g>s"	""
		标签		<CTRL-g>S<html>	<html> </html>
字符可视化模式	选中的文本	括号	print var1, var2	veeS)	print (var1, var2)
行可视化模式	选中的文本	标签	The passion to save humanity is a cover for the desire to rule it.	vjS<p>	<p>The passion to save humanity is a cover for the desire to rule it.</p>
块可视化模式	选中的文本	标签	Item 1 Item 2 Item 3	<Ctrl-Q>\$jjs<li>	<li>Item 1</li> <li>Item 2</li> <li>Item 3</li>

请注意：

1. 插入模式下的操作：
  - 首先同时按下 **CTRL-g** 键；
  - 然后松开 **CTRL** 键；
  - 最后点击 **s** 或 **S** 键，并输入环绕字符或标签。
2. 可视化模式下的操作：
  - 首先使用快捷键进入不同类型的可视化模式。  
比如在Windows下，使用 **CTRL-Q** 键进入块视化模式；
  - 然后使用 **j** 等移动命令来选中文本；
  - 最后点击 **S** 键，并输入环绕字符或标签。

### 修改环绕字符

使用 **cs** 命令可以修改环绕字符：

目标范围	替换内容	原始文本	命令	更改效果
单词	引号	"Hello World"	cs" '	'Hello World'
单词	标签及新行	'Hello World'	cs'<p>	<p> Hello World </p>

单词	标签	<p>Hello World</p>	<b>cst&lt;h1&gt;</b>	<h1>Hello World</h1>
单词	引号	Hello World!	<b>csw'</b>	Hello 'World'!
单词及标点	引号	Hello World!	<b>csw'</b>	Hello 'World'!
单词	括号	(1,2,3)	<b>cs)]</b>	[1,2,3]
单词	括号及空格	(1,2,3)	<b>cs)[</b>	[ 1,2,3 ]

## 删除环绕字符

使用 **ds** 命令可以删除环绕字符：

目标范围	新增内容	原始文本	命令	更改效果
单词	引号	'Hello World'	<b>ds'</b>	Hello World
单词	括号	(123+4*56)/2	<b>ds(</b>	123+4*56/2
单词	标签	<div>Hello</div>	<b>dst</b>	Hello

surround插件主要提供以下命令：

<b>ys</b>	添加环绕字符
<b>yS</b>	添加环绕字符并拆分新行
<b>yss</b>	为整行添加环绕字符
<b>ySS</b>	为整行添加环绕字符并拆分新行
<b>cs</b>	修改环绕字符
<b>cS</b>	修改环绕字符并拆分新行
<b>ds</b>	删除环绕字符

surround插件可以识别并处理以下目标实体：

- **( )**
- **{ }**
- **[ ]**
- **< >**
- **`**
- **"**
- **'**
- **t** (标签)
- **w** (单词)

## 自定义快捷键

如果查看surround插件的源码，将会发现快捷键逐一定义在[surround.vim](#)文件中：

```

599  if !exists("g:surround_no_mappings") || ! g:surround_no_mappings
600      nmap ds <Plug>Dsurround
601      nmap cs <Plug>Csurround
602      nmap cS <Plug>CSurround
603      nmap ys <Plug>Ysurround
604      nmap yS <Plug>YSurround
605      nmap yss <Plug>Yssurround
606      nmap ySs <Plug>YSsurround
607      nmap ySS <Plug>YSSurround
608      xmap S <Plug>VSurround
609      xmap gS <Plug>VgSurround
610  if !exists("g:surround_no_insert_mappings") || ! g:surround_no_insert_mappings
611      if !hasmapto("<Plug>Isurround","i") && "" == mapcheck("<C-S>","i")
612          imap <C-S> <Plug>Isurround
613      endif
614      imap <C-G>s <Plug>Isurround
615      imap <C-G>S <Plug>ISurround
616  endif
617  endif

```

也就是说，我们可以直接在vimrc配置文件中，重置或取消插件预定义的快捷键：

```

" 不定义任何快捷键
let g:surround_no_mappings = 1
" 从plugin/surround.vim复制快捷键定义
" 并按需要进行修改
nmap ds <Plug>Dsurround
nmap cs <Plug>Csurround
nmap cS <Plug>CSurround
nmap ys <Plug>Ysurround
nmap yS <Plug>YSurround
nmap yss <Plug>Yssurround
nmap ySs <Plug>YSsurround
nmap ySS <Plug>YSSurround
" 取消不需要的快捷键
"xmap S <Plug>VSurround
"xmap gS <Plug>VgSurround
"imap <C-S> <Plug>Isurround
imap <C-G>s <Plug>Isurround
imap <C-G>S <Plug>ISurround

```

假设需要输入以下包含多个环绕字符的文本：

```
{{ nginx_root }}
```

由于surround插件并不支持.重复命令，如果想要新增多个成对字符，那么需要安装额外的[repeat.vim](#)插件。而死板繁琐的替代方案是，自定义快捷键来重复执行命令：

```
nmap <C-J> ysiw}lysiw{
```

使用以下命令，可以查看插件的帮助文件：

```
:help surround
```

## 使用感受

---

一，舍本逐末。为了输入环绕字符的小需求，而发明碾压一切的大轮子。颇有些杀鸡用牛刀的意味。而且平白多出来的快捷键和命令，不但冗长而且别扭，操作起来总有种有苦难言的感觉。

二，盛名之下，其实难副。对于一个被不断被提及的知名插件，从功能和体验上都差强人意，着实让人失望。也许，这就是见面不如闻名吧。

以上，皆为一家之言，还请自行斟酌。

---

<[上一篇](#) | [目录](#) | [下一篇](#)>

Ver: 2.0 | [YYQ](#)