

# 七夕节，在Vim中找到你的"另一半"

## g 用作一对通用对象

如果你熟悉 vim 中的文本对象(凑成一对的字符)，那么你一定习惯使用这些 `ci(`、`di{`、`ci"`、`ya[` 组合键，但是，有时你可能会发现这需要一点时间来确定按哪个键，更不用说 `{<"` 距离手指有点距离。这就是 gmotion 可以帮助你地方，而不是输入 `i(`、`i{`、`a[`、`a<`，输入 `ig` 和 `ag` 将起到相同的作用。同时使用突出显示的前缀和后缀，您可以轻松判断哪个文本对象可以被操作。

ig 和 ag 样例:



```
4 |      finish
3 |
2 | g:loaded_gmotion = 1
1 |
16 | "txt"      'txt' {txt}
1 | <txt>      {txt} (txt)
2 | (((((( (((((( (((((( (((((( (((((( (((((( (((((( ((((((
3 |
4 | const g:pair = [
5 |     ['(', ')'],
6 |     ['[', ']'],
7 |     ['{', '}'],
8 |     ['<', '>']
```

## gh gl 找到另一半

通过输入“gh”或“gl”，您可以快速移动到文本对象的左侧或右侧部分。

```
5 |  
4 g:loaded_gmotion = 1  
3 const g:pair = [  
2 |   ['(', ')'],  
1 |   ['[', ''],  
18 |   ['{', '}'],  
1 |   ['<', '>'],  
2 |   ['"', '"'],  
3 |   ['"', '"'],  
4 | ]
```

# Super+r

## 快速拆散、重组一对

gmotion 还可以帮助您更改或删除文本对象的前缀和后缀。

```
4 |   ['"', '"'],  
3 |   ['"', '"'],  
2 | ]  
1 |  
28 | # char to string  
1 |   'char'  
2 | # array to list  
3 |   {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}  
4 | # list to param  
5 |   func([param1, param2, param2])  
6 |  
7 |  
8 |  
9 | const lpart = g:pair→copy()→map((f, i)→
```

# 自定义配置

## 添加您的文本对象

通过更改 `.vimrc` 中的 `g:gmotion_pair`，您可以添加或删除自定义的文本对象，

```
1 let g:gmotion_pair = [  
2     \ ['(', ')'],  
3     \ ['[', ']'],  
4     \ ['{', '}'],  
5     \ ['<', '>'],  
6     \ ['"', '"'],  
7     \ ['"', '"'],  
8     \ ['`', '`']]
```

## 更改高亮组

您还可以自定义高亮组 `gmotion` 使用，改变 `g:gmotion_highligh_group` 来指定使用哪个高亮组

```
1 let g:gmotion_highligh_group = 'MatchParen'
```

## 怎么安装

### 对于 Vim-Plug 管理器

在 `.vimrc` 的插件部分添加以下行

```
1 Plug 'Chen-Rong-Zi/gmotion.vim', { 'do': { -> popup_notification(['Gmotion  
Installed'], {'time': 1000})} }
```

`source` 你的 `.vimrc` 和 `:PlugInstall`，你可能会注意到弹出的通知窗口显示 `Gmotion Installed`

### 如果你不使用任何插件管理器

```
1 cp ./gmotion/plugin/gmoiton.vim $HOME/.vim/plugin/
```

`source` 你的 `.vimrc`