<https://github.com/viruschidai/vim-as-nodejs-ide>

# [viruschidai](https://github.com/viruschidai)/[vim-as-nodejs-ide](https://github.com/viruschidai/vim-as-nodejs-ide)

通过鼠标在centos的terminal中的vim程序copy操作（可能有误，以实践为准）：

普通copy: shift + left mouse cursor drag -> right click mouse cursor -> copy menu

块block矩形区域copy: shift + ctrl + left mouse cursor drag -> right click mouse cursor -> copy menu

（注：在centos的terminal中的vim窗口中，如果copy到其他地方后，记得用相同的按键+鼠标键再点击一下来取消选中）

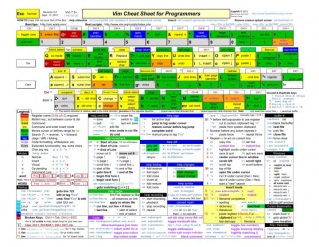
windows系统putty中vim的鼠标copy：

普通copy: shift + left mouse cursor drag

块block矩形区域copy: alt + left mouse cursor drag

<https://rumorscity.com/2014/08/16/5-best-vim-cheat-sheet/>

<http://michael.peopleofhonoronly.com/vim/>



<http://www.catswhocode.com/blog/100-vim-commands-every-programmer-should-know>

## [100 Vim commands every programmer should know](http://www.catswhocode.com/blog/100-vim-commands-every-programmer-should-know)

vundle

ctrlp.vim

fugitive

jedi-vim

jshint2.vim

matchit

minibufexpl (minibufexpl解决了minibufexplore的一些bug)

nerdcommenter

nerdtree

plugin

rope

# simple-pairs vs [jiangmiao](https://github.com/jiangmiao)/[auto-pairs](https://github.com/jiangmiao/auto-pairs)（auto-pairs在插入单独的括号等符号时很不灵活，对其他某些操作有影响，所以应该选择使用simple-pairs）

sparkup

syntastic

tabular

tagbar

ultisnips

vim-airline

vim-coffee-script

vim-easymotion

vim-indent-guides

vim-jade

vim-multiple-cursors

vim-repeat

vim-scala

vim-stylus

vim-surround

YouCompleteMe

<http://stackoverflow.com/questions/6787734/strange-behavior-of-vim-color-inside-screen-with-256-colors>

# [Strange behavior of vim color inside screen with 256 colors](http://stackoverflow.com/questions/6787734/strange-behavior-of-vim-color-inside-screen-with-256-colors)

如下的配置.screenrc可以解决vim在screen中 颜色不一致的问题。

Three years later after I asked this question, today I saw a similar problem. The problem is that vimcan display 256 colors outside screen, and screen can display 256 colors with a test script, but vim can't display any color (can only display black and white) inside screen. Just as a note to myself, here is the .screenrc file I am using

hardstatus alwayslastline "%{.bW}%-w%{.rW}%n %t%{-}%+w %=%{..G} %H %{..Y} %Y-%m-%d %c"

shell "bash"

startup\_message off

vbell off

altscreen on

attrcolor b ".I"

defbce "on"

termcapinfo xterm\* 'is=\E[r\E[m\E[2J\E[H\E[?7h\E[?1;4;6l'

termcapinfo xterm 'Co#256:AB=\E[48;5;%dm:AF=\E[38;5;%dm'

term screen-256color

The solution to the problem is already mentioned in the accepted answer, namely, I need to include

export TERM=xterm-256color

in .bashrc.

解决vim在screen中不能用鼠标拖动来改变split window大小的问题。

<http://superuser.com/questions/549930/cant-resize-vim-splits-inside-tmux>

It appears that dragging the status line to resize a split is not possible when the *Vim* option ttymouseis xterm; it does work when the value is xterm2 though. The latter value configures Vim to ask for an extended mouse reporting mode that (among other things) provides better dragging support. This extended mode only works with newer versions of *xterm* (and other compatible terminal emulators, including *tmux*), so it is not the default value.

You could use something like the following in your .vimrc to set the option: （本人采用的是这种方法）

set mouse+=a

if &term =~ '^screen'

" tmux knows the extended mouse mode

set ttymouse=xterm2

endif

(Though, I am not sure how this will affect actual *screen* instances, which also use a TERM that starts with screen.)

When you are outside of *tmux*, the TERM environment variable is probably an xterm-ish value, and*Vim* will probe for the *xterm* version by using the t\_RV control sequence.

# [Can't resize vim splits inside tmux](http://superuser.com/questions/549930/cant-resize-vim-splits-inside-tmux)

In my case it resolved both cases: mouse split resize and mouse position problem for wide screen.

[The fix is:](http://stackoverflow.com/questions/7000960/vim-mouse-problem)

if has("mouse\_sgr")

set ttymouse=sgr

else

set ttymouse=xterm2

end

# [In Vim, why doesn't my mouse work past the 220th column?](http://stackoverflow.com/questions/7000960/in-vim-why-doesnt-my-mouse-work-past-the-220th-column)

This has been fixed in [Vim 7.3.632](ftp://ftp.vim.org/pub/vim/patches/7.3/7.3.632). See :h sgr-mouse. Or just put this in your ~/.vimrc:

set ttymouse=sgr

If you want to be compatible with versions that don't have mouse\_sgr compiled in, use:

if has("mouse\_sgr")

set ttymouse=sgr

else

set ttymouse=xterm2

end

To see if your version of Vim has mouse\_sgr, run vim --version from the command-line, or in Vim, enter :version, and look for +mouse\_sgr.

[**https://github.com/suan/vim-instant-markdown**](https://github.com/suan/vim-instant-markdown)

# [suan](https://github.com/suan)/[vim-instant-markdown](https://github.com/suan/vim-instant-markdown)

# vim-instant-markdown

Want to instantly preview finnicky markdown files, but don't want to leave your favorite editor, or have to do it in some crappy browser textarea? **vim-instant-markdown** is your friend! When you open a markdown file in vim, a browser window will open which shows the compiled markdown in real-time, and closes once you close the file in vim.

As a bonus, [github-flavored-markdown](http://github.github.com/github-flavored-markdown/) is supported, and styles used while previewing are the same as those github uses!

[**https://en.wikipedia.org/wiki/Markdown**](https://en.wikipedia.org/wiki/Markdown)

[**https://github.com/plasticboy/vim-markdown**](https://github.com/plasticboy/vim-markdown)

<http://vimawesome.com/>



我已有的插件：



<https://github.com/tpope/vim-surround>

# [tpope](https://github.com/tpope)/[vim-surround](https://github.com/tpope/vim-surround)

# surround.vim

Surround.vim is all about "surroundings": parentheses, brackets, quotes, XML tags, and more. The plugin provides mappings to easily delete, change and add such surroundings in pairs.

It's easiest to explain with examples. Press **cs"'** inside

"Hello world!"

to change it to

'Hello world!'

Now press cs'<q> to change it to

<q>Hello world!</q>

To go full circle, press cst" to get **联想：cs + t [c表示create, s表示surround, t表示tag]**

"Hello world!"

To remove the delimiters entirely, press ds". **联想：delete sourround**

Hello world!

Now with the cursor on "Hello", press ysiw] (iw is a text object). **联想：[yard sourround in word]**

[Hello] world!

Let's make that braces and add some space (use } instead of { for no space): cs]{

{ Hello } world!

Now wrap the entire line in parentheses with yssb or yss).

({ Hello } world!)

Revert to the original text: ds{ds)

Hello world!

Emphasize hello: ysiw<em>

<em>Hello</em> world!

Finally, let's try out visual mode. Press a capital V (for linewise visual mode) followed by S<p class="important">.

<p class="important">

<em>Hello</em> world!

</p>

This plugin is very powerful for HTML and XML editing, a niche which currently seems underfilled in Vim land. (As opposed to HTML/XML inserting, for which many plugins are available). Adding, changing, and removing pairs of tags simultaneously is a breeze.

The . command will work with ds, cs, and yss if you install [**repeat.vim**](https://github.com/tpope/vim-repeat)**.**

## Installation

If you don't have a preferred installation method, I recommend installing [pathogen.vim](https://github.com/tpope/vim-pathogen), and then simply copy and paste:

cd ~/.vim/bundle

git clone git://github.com/tpope/vim-surround.git

Once help tags have been generated, you can view the manual with :help surround.

<http://www.catonmat.net/blog/vim-plugins-repeat-vim/>

# [Vim Plugins You Should Know About, Part II: repeat.vim](http://www.catonmat.net/blog/vim-plugins-repeat-vim/)

Here comes the second post in the article series "**Vim Plugins You Should Know About**". This time I am going to introduce you to a plugin called "**repeat.vim**".

Repeat.vim fixes an important functionality problem in the [surround.vim plugin](http://www.catonmat.net/blog/vim-plugins-surround-vim/) that I wrote about last week. The problem with surround.vim lies in the repeat command "." (dot). If you had applied a surrounding and wanted to repeat it with the "." command, it wouldn't work. This plugin fixes this problem.

So basically, **whenever you install surround.vim, you also want to install repeat.vim with it**.

There is one catch, though. It does not repeat visual or "ys" commands. Luckily, the "ys" commands can be often be substituted with "cs" commands. For example, if you wanted to do several "ysw"" (wrap a word in quotes), you may type "csw"" and then use "." commands to repeat. The only way to repeat visual commands is to record a macro.

Here is an example usage of the repeat.vim script. Suppose you had typed a sentence and you wanted to wrap all the words in quotes:

**|**foo bar baz quux muux woox

(**|** is cursor)

Type **csw"**:

**|**"foo" bar baz quux muux woox

Now press **W.**

"foo" **|**"bar" baz quux muux woox

**W** moved to the next word and **.** repeated the wrapping command.

Now do the same 4 more times, and you have the whole line wrapped:

"foo" "bar" "baz" "quux" "muux" "woox"

## How to install repeat.vim?

* 1. Download [repeat.vim](http://www.vim.org/scripts/script.php?script_id=2136) to ~/.vim/plugin (on Unix/Linux), or ~\vimfiles\plugin (Windows).
* 2. Restart Vim or source repeat.vim with ":so ~/.vim/plugin/repeat.vim" on Unix or ":so ~/vimfiles/plugin/repeat.vim" on Windows).

## Have Fun!

Have fun with surround.vim + repeat.vim. I'll write about a much more life-changing plugin the next time. :)

<http://www.catonmat.net/blog/vim-plugins-matchit-vim/>

# [Vim Plugins You Should Know About, Part III: matchit.vim](http://www.catonmat.net/blog/vim-plugins-matchit-vim/)

# <https://github.com/tmhedberg/matchit>

# [tmhedberg](https://github.com/tmhedberg)/[matchit](https://github.com/tmhedberg/matchit)

<http://www.catonmat.net/series/vim-plugins-you-should-know-about>

<http://www.bestofvim.com/plugin/>

<http://nvie.com/posts/how-i-boosted-my-vim/>

# How I boosted my Vim

<http://vim.wikia.com/wiki/Easier_buffer_switching>

## Conveniently accessing buffers[Edit](http://vim.wikia.com/wiki/Easier_buffer_switching?action=edit&section=8)

### Built-in features[Edit](http://vim.wikia.com/wiki/Easier_buffer_switching?action=edit&section=9)

In gvim, you can use the Buffers menu to conveniently access buffers (tear off the menu to make an always-visible list).

Or, put the following in your [vimrc](http://vim.wikia.com/wiki/Vimrc):

set wildchar=<Tab> wildmenu wildmode=full

Now, pressing Tab on the command line will show a menu to complete buffer and file names. If you include the following, you can also press F10 to open the buffer menu.

set wildcharm=<C-Z>

nnoremap <F10> :b <C-Z>

Try this example to see how powerful the wildmenu is.

:args vehicle.c vehicle.h car.c car.h jet.c jet.h jetcar.c jetcar.h

:b <Tab> " offers all buffers in a menu

:b car<Tab> " offers car.c car.h

:b \*car<Tab> " offers car.c jetcar.c car.h jetcar.h

:b .h<Tab> " offers the \*.h buffers

:b .c<Tab> " offers the \*.c buffers

:b ar.c<Tab> " offers car.c jetcar.c

:b j\*c<Tab> " offers jet.c jetcar.c jetcar.h

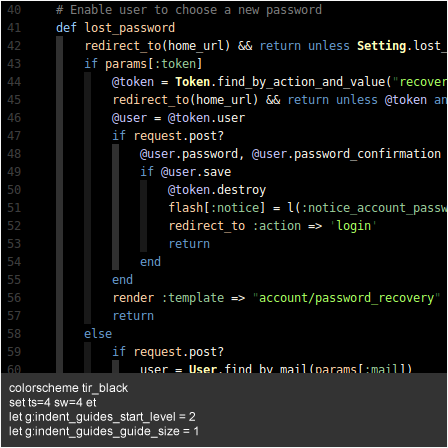
[**https://github.com/nathanaelkane/vim-indent-guides**](https://github.com/nathanaelkane/vim-indent-guides)

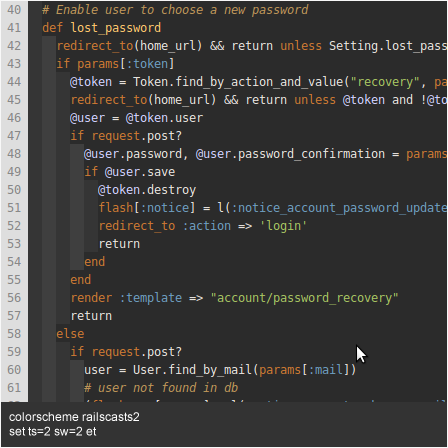
[nathanaelkane](https://github.com/nathanaelkane)/[vim-indent-guides](https://github.com/nathanaelkane/vim-indent-guides)

let g:indent\_guides\_auto\_colors = 0

autocmd VimEnter,Colorscheme \* :hi IndentGuidesOdd guibg=red ctermbg=3

autocmd VimEnter,Colorscheme \* :hi IndentGuidesEven guibg=green ctermbg=4

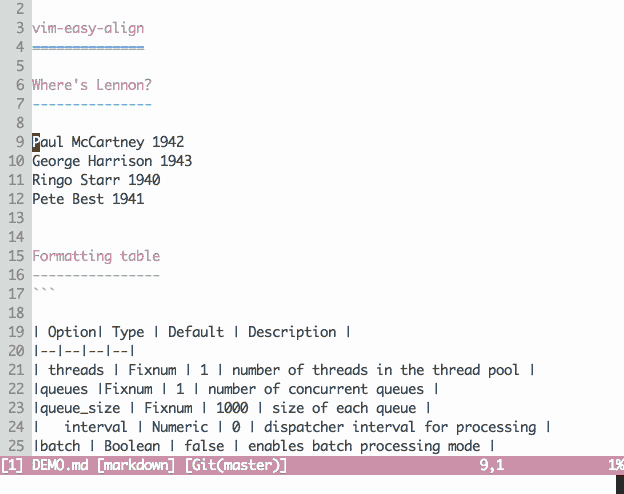




[How do I get Vim to automatically put ending braces?](http://stackoverflow.com/questions/883437/how-do-i-get-vim-to-automatically-put-ending-braces)

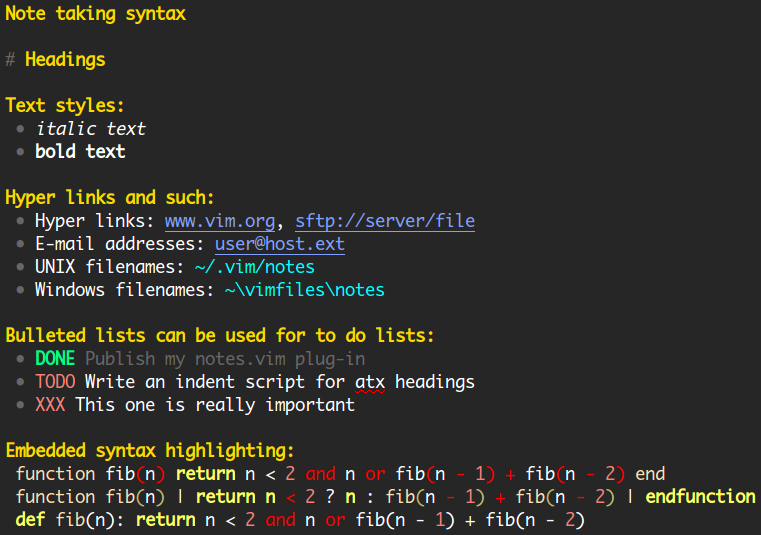
<https://github.com/junegunn/vim-easy-align>

[junegunn](https://github.com/junegunn)/[**vim-easy-align**](https://github.com/junegunn/vim-easy-align)



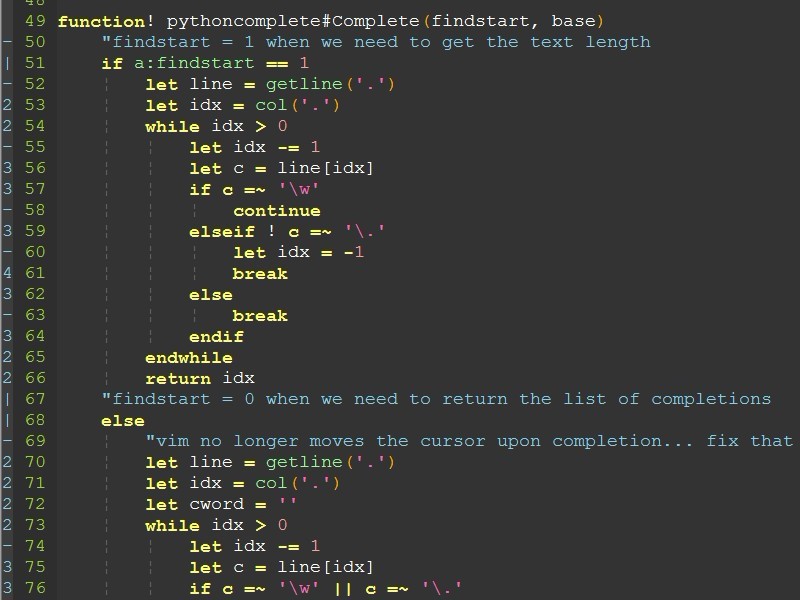
<https://github.com/xolox/vim-notes>

# [xolox](https://github.com/xolox)/[vim-notes](https://github.com/xolox/vim-notes)



<https://github.com/Yggdroot/indentLine>

# [Yggdroot](https://github.com/Yggdroot)/[indentLine](https://github.com/Yggdroot/indentLine)



# [Vim Plugins You Should Know About, Part IV: snipmate.vim](http://www.catonmat.net/blog/vim-plugins-snipmate-vim/)

# <http://www.catonmat.net/series/vim-plugins-you-should-know-about>

# [Article series "Vim Plugins You Should Know About"](http://www.catonmat.net/series/vim-plugins-you-should-know-about)

Article series description:

In this article series I explain various Vim text editor plugins you should know about. They make you much more productive!

Article series has **7** posts:

[Vim Plugins You Should Know About, Part I: surround.vim](http://www.catonmat.net/blog/vim-plugins-surround-vim/)

**37** comments

**484,936** views

December 08, 2008

[Vim Plugins You Should Know About, Part II: repeat.vim](http://www.catonmat.net/blog/vim-plugins-repeat-vim/)

**13** comments

**90,693** views

December 15, 2008

[Vim Plugins You Should Know About, Part III: matchit.vim](http://www.catonmat.net/blog/vim-plugins-matchit-vim/)

**13** comments

**212,516** views

February 05, 2009

[Vim Plugins You Should Know About, Part IV: snipmate.vim](http://www.catonmat.net/blog/vim-plugins-snipmate-vim/)

**36** comments

**210,761** views

August 04, 2009

[Vim Plugins You Should Know About, Part VI: nerd\_tree.vim](http://www.catonmat.net/blog/vim-plugins-nerdtree-vim/)

**21** comments

**435,851** views

January 18, 2010

[Vim Plugins You Should Know About, Part V: a.vim](http://www.catonmat.net/blog/vim-plugins-a-vim/)

**10** comments

**75,519** views

November 05, 2009

[Vim Plugins You Should Know About, Part VII: ragtag.vim (formerly allml.vim)](http://www.catonmat.net/blog/vim-plugins-ragtag-allml-vim/)

**12** comments

**155,718** views

March 03, 2010

Article series statistics:

<http://benmccormick.org/2014/07/21/learning-vim-in-2014-getting-more-from-vim-with-plugins/>

# [Learning Vim in 2014: Getting More from Vim with Plugins](http://benmccormick.org/2014/07/21/learning-vim-in-2014-getting-more-from-vim-with-plugins/)

<https://github.com/nodejs/node-v0.x-archive/wiki/Vim-Plugins>

# [nodejs](https://github.com/nodejs)/[node-v0.x-archive](https://github.com/nodejs/node-v0.x-archive)

<http://code.tutsplus.com/series/vim-essential-plugins--net-19224>

# Vim Essential Plugins

8 Posts

<https://github.com/rstacruz/sparkup>

# [rstacruz](https://github.com/rstacruz)/[sparkup](https://github.com/rstacruz/sparkup)

* **VIM**: See the vim/README.txt file for installation. In VIM, create or open an HTML file (or set the filetype to html), type in something (e.g. #header > h1), then press <C-E> whilst in **insert mode**to expand to HTML. Pressing <C-n> will cycle through empty elements. Variables specified invim/README.txt can be used to customise key mappings, and to add **normal mode** mappings as well.

## Examples

**div** expands to:

<div></div>

**div#header** expands to:

<div id="header"></div>

**div.align-left#header** expands to:

<div id="header" class="align-left"></div>

**div#header + div#footer** expands to:

<div id="header"></div>

<div id="footer"></div>

**#menu > ul** expands to:

<div id="menu">

<ul></ul>

</div>

**#menu > h3 + ul** expands to:

<div id="menu">

<h3></h3>

<ul></ul>

</div>

**#header > h1{Welcome to our site}** expands to:

<div id="header">

<h1>Welcome to our site</h1>

</div>

**a[href=index.html]{Home}** expands to:

<a href="index.html">Home</a>

**ul > li\*3** expands to:

<ul>

<li></li>

<li></li>

<li></li>

</ul>

**ul > li.item-$\*3** expands to:

<ul>

<li class="item-1"></li>

<li class="item-2"></li>

<li class="item-3"></li>

</ul>

**ul > li.item-$\*3 > strong** expands to:

<ul>

<li class="item-1"><strong></strong></li>

<li class="item-2"><strong></strong></li>

<li class="item-3"><strong></strong></li>

</ul>

**table > tr\*2 > td.name + td\*3** expands to:

<table>

<tr>

<td class="name"></td>

<td></td>

<td></td>

<td></td>

</tr>

<tr>

<td class="name"></td>

<td></td>

<td></td>

<td></td>

</tr>

</table>

**#header > ul > li < p{Footer}** expands to:

<!-- The < symbol goes back up the parent; i.e., the opposite of >. -->

<div id="header">

<ul>

<li></li>

</ul>

<p>Footer</p>

</div>

<http://blog.csdn.net/lihui130135/article/details/44619309>

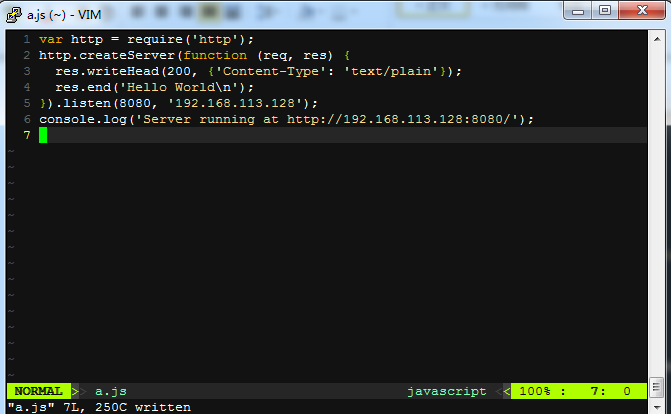
[vim 高级使用技巧第一篇](http://blog.csdn.net/lihui130135/article/details/44619309)

<http://www.zlovezl.cn/articles/my-vim-plugins-for-python/>

# 我常用的Vim插件(python向)

<https://github.com/sjl/badwolf>

# [sjl](https://github.com/sjl)/[badwolf](https://github.com/sjl/badwolf)



# [mattn](https://github.com/mattn)/[emmet-vim](https://github.com/mattn/emmet-vim)

<https://github.com/mattn/emmet-vim>

