

```
601 " @return [pattern, offset]
602 function! incsearch#parse_pattern(expr, search_k
603     " search_key : '/' or '?'
604     " expr       : {pattern}/{pattern}/{offset}
605     " expr       : {pattern}/;/{newpattern} :h /
606     " return     : [{pattern}/{pattern}, {offset}]
607     let very_magic = '\v'
608     let pattern = '(%(\| )[-])'
609     let slash = '[:blank:]' . '&[^\\"]|[:
610     let offset = '(.*)'
611
612     let parse_pattern = very_magic . pattern . '
        slash . offset . ')?$'
incsearch.vim | dev
```

/-improved

/pattern 2014/11/08 (Sat) #vimconf2014 @haya14busa



hayal4busa



@hayal4busa



@hayal4busa



/-Search

/ as a Search

```
1 " haya14busa's vimrc
2 " Author: haya14busa
3 " URL: http://haya14busa.com
4 " Source: https://github.com/haya14busa/dotfiles/
5 "=====
6 "
7 "      _/ /_      _/ / / / / / / /_
8 "      / _/ \_ / _/ / / / / _/ \_ /
9 "      / / / / / / / / / / / / / / ( _ ) / / /
10 "     / _/ \_ / \_ / \_ / \_ / \_ / \_ / \_ / \_ /
11 "              / _/ /
12 "
13 "=====
14
15 " Startup {{{=====
16 " Note: Skip initialization for vim-tiny or vim-small.
17 if !1 | finish | endif
```

/ as a Search

/ as a Motion

/ as a Motion

```
518 " Colorscheme {{{
519 " Check color
520 " :so $VIMRUNTIME/syntax/colortest.vim
521 if has('vim_starting')
522     syntax enable
523     set background=dark
524     set t_Co=256
525     if &t_Co < 256
526         colorscheme default
527     else
528         try
529             colorscheme molokai
530         catch
531             colorscheme desert
532         endtry
533     endif
534 endif
```

Repeat: `:h last-pattern`

Completion: `:h c_ <C-r> _ <C-w>`

Command: `:h {offset}`

Regular Expression


```
519 " Check color
520 " :so $VIMRUNTIME/syntax/colortest.vim
521 if has('vim_starting')
522     syntax enable
523     set background=dark
524     set t_Co=256
525     if &t_Co < 256
526         | colorscheme default
527     else
528         | try
529         | | colorscheme molokai
530         | catch
531         | | colorscheme blue
532         | endtry
533     endif
534 endif
535 "}}}}
```

/ as a Motion

completion: c_<C-r>_<C-w>

offset: //e & last-pattern

regex: `` etc...

/with Operator

:h gn

```

674
675 function! incsearch#detect_case(pattern)
676     " Ignore \%C, \%U, \%V for smartcase detection
677     let pattern = substitute(a:pattern, s:non_escaped_backslash . '%[CUV]', '', 'g')
678     " Explicit \c has highest priority
679     if pattern =~# s:non_escaped_backslash . 'c'
680         return '\c'
681     endif
682     if pattern =~# s:non_escaped_backslash . 'C' || &ignorecase == s:FALSE
683         return '\C' " noignorecase or explicit \C
684     endif
685     if &smartcase == s:FALSE
686         return '\c' " ignorecase & nosmartcase
687     endif
688     " Find uppercase letter which isn't escaped
689     if pattern =~# s:escaped_backslash . '[A-Z]'
690         return '\C' " smartcase with [A-Z]
691     else
692         return '\c' " smartcase without [A-Z]
693     endif
694 endfunction
695
696 function! incsearch#convert_with_case(pattern)
697     return incsearch#detect_case(a:pattern) . a:pattern
698 endfunction
699

```

gn

/ & {operator}gn & . (dot repeat)

ok, / is benri



USEFUL

benri \doteq useful

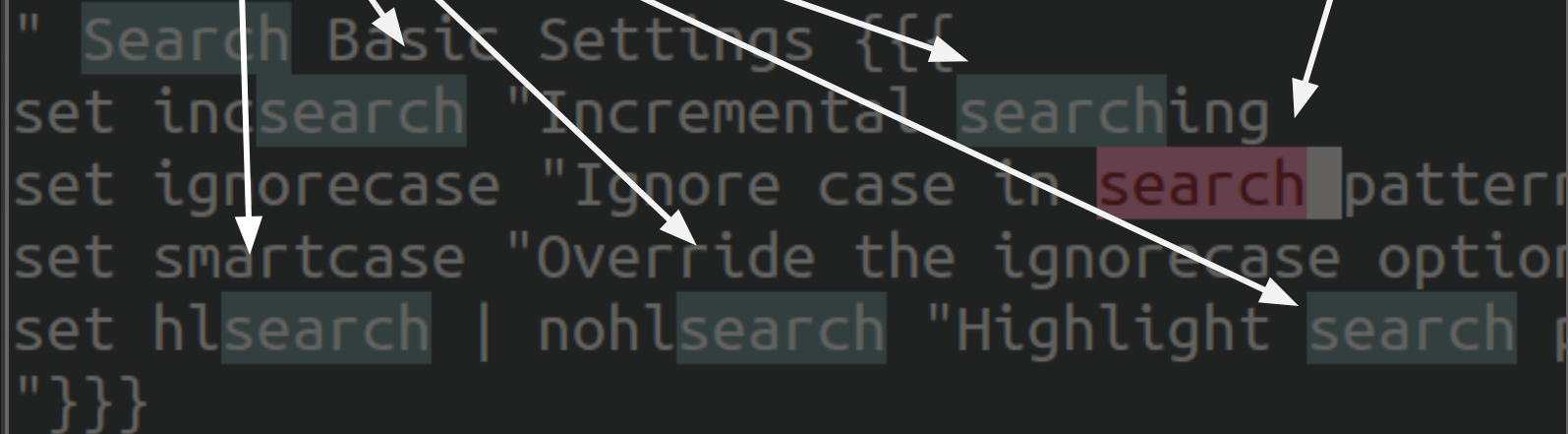
Emacs

```
" Search Basic Settings {{{  
set incsearch "Incremental searching  
set ignorecase "Ignore case in search pattern  
set smartcase "Override the ignorecase option  
set hlsearch | nohlsearch "Highlight search  
"}}}
```

Highlight all matches

Emacs

*Incremental cursor
moving*



The diagram shows a screenshot of Emacs search settings with several annotations. A central point from which five arrows originate is located above the first line of code. The arrows point to the following elements: the word 'Search', the word 'Incremental', the word 'search' in the 'Ignore case' setting, the word 'Override' in the 'smartcase' setting, and the word 'Highlight' in the 'hlsearch' setting. A separate arrow points from the text 'Incremental cursor moving' to the 'Incremental' setting.

```
" Search Basic Settings {{{  
set incsearch "Incremental searching  
set ignorecase "Ignore case in search pattern  
set smartcase "Override the ignorecase option  
set hlsearch | nohlsearch "Highlight search p  
"}}}
```




/ is *not benri* enough...

benri \doteq useful



Solution



hayal4buser/incsearch.vim

```

251 function! incsearch#parse_pattern(expr, search_key)
252     " search_key : '/'
253     " expr       : /{pattern}/{pattern}/{flags}
254     " return     : [{pattern}/{pattern}, {flags}]
255     let very_magic = '\v'
256     let pattern = '(%(\\\|.|.){-})'
257     let slash = '(\'' . a:search_key . '&[^\\\"|[:alnum:][:blank:]])'
258     let flags = '(.*)'
259
260     let parse_pattern = very_magic . pattern . '(%(' . slash . flags . ')?$'
261     let result = matchlist(a:expr, parse_pattern)[1:3]
262     if type(result) == type(0) || empty(result)
263         | return []
264     endif
265     unlet result[1]
266     return result
267 endfunction
268
269 function! incsearch#convert(pattern)
270     if &ignorecase == s:FALSE
271         | return '\C' . a:pattern " noignorecase

```

haya14busa/incsearch.vim

50
51
52
53
54
55
56
57
58

```
" \v  
" \\v  
" \\\v  
" \\\\v  
" \\\\v  
" \\\\v  
" \\\\v  
" \v\v\v\v
```

Highlight all matched patterns

Interactive Regular expression editing

```
1 "=====
2 " FILE: autoload/incsearch.vim
3 " AUTHOR: haya14busa
4 " License: MIT license {{{
5 "     Permission is hereby granted, free of charge, to any person obtaining
6 "     a copy of this software and associated documentation files (the "
7 "     "Software"), to deal in the Software without restriction, including
8 "     without limitation the rights to use, copy, modify, merge, publish,
9 "     distribute, sublicense, and/or sell copies of the Software, and to
10 "    permit persons to whom the Software is furnished to do so, subject to
11 "    the following conditions:
12 "
13 "    The above copyright notice and this permission notice shall be included
14 "    in all copies or substantial portions of the Software.
15 "
16 "    THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
17 "    OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
18 "    MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
19 "    IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
20 "    CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
21 "    TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
```

*Incremental cursor moving
No need to press nnnnnn...*

Design

- 1. Simple*
- 2. Comfortable*
- 3. Useful*

Minimal Installation

```
1 NeoBundle "haya14busa/incsearch.vim"  
2 Plugin "haya14busa/incsearch.vim"  
3 Plug "haya14busa/incsearch.vim"  
4  
5 map / <Plug>(incsearch-forward)  
6 map ? <Plug>(incsearch-backward)
```

See :h incsearch.txt

Development

Use lessons learned from past my experiences

Lessons from Development of vim-easymotion

Vimプラグイン読書会(reading-vimplugin)

- The way to handle dot repeat with <expr> mapping*

Thank you everyone

Command Line Interface

- *osyo-manga/vital-over (special thanks @osyo-manga)*
 - *I request lots of feature and he developed it with his awesome skill! ;)*
- *(vim-over, easymotion, incsearch.vim)*

I highly appreciate him, thank you @osyo-manga :)

Thank you everyone

Regular expression

- Parse command-line with modified regex from osyomanga/over.vim, which is based on @thinca's work
- Handle escaped backslash(\) with little-improved regex from deris/vim-magicalize and incsearch.vim use this everywhere now!

219 [✓] repetition_backward_change

220 [✓] repetition_stay_change

221 incremental_nextip

222 [✓] inc_next_forward

223 [✓] inc_next_backward

224 [✓] inc_prev_forward

225 [✓] inc_prev_backward

226 - thinca/vim-themis.vim

227 [✓] inc_prev_stay

228 [✓] inc_last_pattern

229 [✓] inc_last_pattern_offset

230 [-] inc_last_pattern_reset

231 Skip because Segmentation fault occurs with histdel()

232

233 tests 129

234 passes 124

235 pendings 5

Thank you everyone

Testing framework

Thank you everyone

vimproc, C/Migemo (, PM2) (in the future)

- *incsearch.vim currently doesn't use these plugins right now, but I developed with these plugins in my development branch and try to provide converter feature*
- *The developers are Yukihiro Nakadaira-san, Shougo-san, Kaoriya-san, Ujihisa-san respectively*

Thank you everyone

Feedback

- [hayal4busa/incsearch.vim/issues](https://github.com/hayal4busa/incsearch.vim/issues)
- Request from users in Lingr
- `c_<C-r>_<C-w>`, missing default feature, bugfix etc...

Thank you everyone

vim-jp/issues

- I bumped into several bugs or strange behavior of vim
- I reported them to vim-jp/issues and vim-jp members kindly made patches ! I respect them continuous works!

Thank you everyone!

yeah!

All stars on github repository and good feedbacks are thanks to everyone ! I'm really glad to develop this plugin and hear good responses from a lot of users.

I'm confident with incsearch.vim, if you are interested in, please give it a shot! I hope you like it ;)

Thank you for listening!

Happy Vim'ing! ♥ @hayal4busa