

Vi IMproved - VIM

Kaylani Bochie

Conteúdo

1. Motivação
2. Demonstração e uso incorreto
3. Modos de operação
4. Operadores e objetos
5. Movimentos
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins

1. Motivação
2. Demonstração e uso incorreto
3. Modos de operação
4. Operadores e objetos
5. Movimentos
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins

Motivação

Usar:

- Onipresente;
- CLI;
- Eficiente e efetivo;
- Editor modal;
- Customizável.

Não usar:

- Falta de componentes *out-of-the-box*;
- Curva de aprendizado.

1. Motivação
2. Demonstração e uso incorreto
3. Modos de operação
4. Operadores e objetos
5. Movimentos
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins

Demonstração e uso incorreto



"Your problem with Vim is that you don't grok vi." - Jim Dennis

1. Motivação
2. Demonstração e uso incorreto
- 3. Modos de operação**
4. Operadores e objetos
5. Movimentos
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins



Modos de operação

Normal

ESC

Inserção

i, a, A

Comando

;, /

Visual

v, V

1. Motivação
2. Demonstração e uso incorreto
3. Modos de operação
4. Operadores e objetos
5. Movimentos
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins

Operadores e objetos

Comando	Descrição
d	delete/cut
y	yank/copy
c	change
gU	make uppercase
~	swap case
<	shift left
=	indent

Objeto	Descrição
aw	a word
iw	in word
aW	a WORD
iW	in WORD
ap	a paragraph
at	tag block (HTML)

1. Motivação
2. Demonstração e uso incorreto
3. Modos de operação
4. Operadores e objetos
- 5. Movimentos**
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins

Movimentos

Movimento	Descrição
h, j, k, l	left, down, up, right
%	find matching bracket
w	word
f <CHAR>	find next <CHAR> in line
F <CHAR>	find previous <CHAR> in line
/ <STRING>	find next <STRING> in file
\$	end of line
O	beginning of line
b	back a word
_	current line
'w	go to marker w
mw	set marker w

1. Motivação
2. Demonstração e uso incorreto
3. Modos de operação
4. Operadores e objetos
5. Movimentos
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins



[count][operator][motion/object]

Comando	Descrição
gUaW	capitaliza uma palavra
gU\$	capitaliza até o final da linha
4cw	corta 4 palavras e entra em modo de inserção
di)	apaga texto dentro de parênteses
d/<STRING>	apaga texto até próxima ocorrência de <STRING>

Comandos essenciais e registradores (:reg)

Comando	Descrição
^f	page down
^b	page up
^e	scroll down
^y	scroll up
gg	first line of file
G	last line of file

Comando	Descrição
d_ OU dd	delete line
y_ OU yy	yank line
p	paste under cursor
P	paste above cursor
^a	increment one
^x	decrement one
.	repeat command

Comando	Descrição
"adi)	mesmo que di) porém coloca no registrador a
"ap	cola conteúdo do registrador a
^n	autocompleta

1. Motivação
2. Demonstração e uso incorreto
3. Modos de operação
4. Operadores e objetos
5. Movimentos
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins



Macros

Gravar

qa<COMANDOS>

Executar

@qa

Modo de comando

Comando	Descrição
:%s/foo/bar/g	substitui foo por bar
:sp	split horizontal
:vsp	split vertical
:tabnew	abre nova aba
:tabp	aba anterior
:tabn	próxima aba
:e <FILENAME>	edit
:r <FILENAME>	read

1. Motivação
2. Demonstração e uso incorreto
3. Modos de operação
4. Operadores e objetos
5. Movimentos
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins

~/.vimrc



1. Motivação
2. Demonstração e uso incorreto
3. Modos de operação
4. Operadores e objetos
5. Movimentos
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins

Regex

Comando	Descrição
.	any character except new line
\s	whitespace character
\S	non-whitespace character
\d	digit
\D	non-digit
\x	hex digit
\p	printable character
\P	like \p, but excluding digits
\w	word character
\W	non-word character
\l	lowercase character
\u	uppercase character

Regex

Comando	Descrição
*	matches 0 or more of the preceding characters
\+	matches 1 or more of the preceding characters
\=	matches 0 or 1 more of the preceding characters
\{n,m\}	matches from n to m of the preceding characters
\{n\}	matches exactly n times of the preceding characters
\{,m\}	matches at most m of the preceding characters
\{n,\}	matches at least n of of the preceding characters
\{-\}	matches 0 or more of the preceding atom
\{-n,\}	matches at lease or more of the preceding characters

Regex

Comando	Descrição
&	the whole matched pattern
\L	the following characters are made lowercas
\O	the whole matched pattern
\U	the following characters are made uppercase
\1	the matched pattern in the first pair of \(\)
\2	the matched pattern in the second pair of \(\)
\E	end of \U and \L
\e	end of \U and
\r	split line in two at this point
\l	next character made lowercase
~	the previous substitute string
\u	next character made uppercase

1. Motivação
2. Demonstração e uso incorreto
3. Modos de operação
4. Operadores e objetos
5. Movimentos
6. [operator][count][motion/object]
7. Macros
8. ~/.vimrc
9. Regex
10. Plugins

Plugins

- Vundle [1];
- NERDTree [2];
- vim-gitgutter [3];
- vim-indent-guides [4];
- YouCompleteMe [5];
- vim-minimap [6].

Referências I

- [1] V. Vim, “Vundle.”

<https://github.com/VundleVim/Vundle.vim>, 2019.

Acessado em 10/08/2019.

- [2] scrooloose, “Nerdtree.”

<https://github.com/scrooloose/nerdtree>, 2019.

Acessado em 10/08/2019.

- [3] airblade, “vim-gitgutter.”

<https://github.com/airblade/vim-gitgutter>, 2019.

Acessado em 10/08/2019.

- [4] nathanaelkane, “Indent guides.”

<https://github.com/nathanaelkane/vim-indent-guides>,
2019.

Acessado em 10/08/2019.

Referências II

[5] ycm core, “Youcomplete.me.”

<https://github.com/ycm-core/YouCompleteMe>, 2019.

Acessado em 10/08/2019.

[6] severin lemaignan, “vim-minimap.”

<https://github.com/severin-lemaignan/vim-minimap>,
2019.

Acessado em 10/08/2019.

:wq