

Vim Cheatsheet



u/blnd_the-dsgner



**Copying and Sharing is allowed.
Please copy and share it.**

blnd_the-dsgner

< h j k l >

^

∨

Write, Quit and Reload a file



write



quit



write&quit



quit

without writing



reload

from disk

Search in Vim

Useful Options for Searching

Search forward with Slash /,
and backward with Question Mark ?.
Keys n and N are used to navigate
for next/previous match.

:set hlsearch

Highlight founded text

:nohlsearch

temporarily turn of highlighting

:set incsearch

Hightlight as you type

/class

Search forward for word class

?namespace

Search backward for word namespace

/\c list

Escape

Character

Search for word list by ignoring cases.

/\<exact\>

Search by exact matching.

/\/**

Search for /**

/function/@

Search for the word function,
and jump to the end of that word.
/w to jumps to the beginning.


Search a character in a line with "f" and "t"

Typing "f" and a given character in Normal Mode, highlights that character. Meanwhile "t" gives the character before that one we typed.

Example:  Search here using t or f and a character.



Press f
followed by e

 Search here using t or f and a character.




Go to the next
occurrence

Search  here using t or f and a character.



Go to the previous
occurrence

Search here using t or f and a character. 

Example:  Search here using t or f and a character.



Press t
followed by e

 Search here using t or f and a character.



Go to the next
occurrence

Search  here using t or f and a character.



Go to the previous
occurrence

 Search here using t or f and a character.

"F" and "T" works in similar way, except that their are used for backward searching on that line. Both ";" and "," have similar function.

Search with * and # in Vim

Star * and Pound # can be used to search forward and backward for the exact word at or near the cursor.

cursor position
`function move() { ... }`

`move();`

`edit();`

`move();`



Press to search

`function move() { ... }`

`move();`

`edit();`

`move();`



next occurrence

`function move() { ... }`

`move();`

`edit();`

`move();`



previous occurrence

`function move() { ... }`

`move();`

`edit();`

`move();`

`function move() { ... }`

`move();`

`edit();`

`move();`



Press to search

`function move() { ... }`

`move();`

`edit();`

`move();`



next occurrence

`function move() { ... }`

`move();`

`edit();`

`move();`



previous occurrence

`function move() { ... }`

`move();`

`edit();`

`move();`

g* and g# can be used to search for not exact match.

`move() { ... }`

`moveleft() { ... }`

`function moveright() { ... }`

In this example, move, moveleft, moveright will be highlighted because we used g*

Substitute

The substitute command allows us to find and replace text.

Command Mode

`:S/javascript/python/g`

Substitute javascript with python, in a line

The g flag means global
Each occurrence in the line is changed,
rather than just the first

`:%S/css/sass/g`

Substitute in all lines

`:5,12s/dev/production/g`

Substitute from line 5 to line 12

`:.,+2s/bug/feature/g`

Change each for the current line (.) and the two next lines (+2)

`:g/^review/s/needs/done/g`

Change each 'needs' to 'done' in each line starting with 'review'

`:%S/chrome/firefox/gc`

Ask for confirmation first

`:%S/text/Editor/gi`

Ignore Cases

`:%S/\<static\>/dynamic/g`

Change only whole words exactly matching

`:%S/remember/me/&`

Reuse last arguments

`:%S/remember/me/n`

This command doesn't substitute anything.
It tells us the number of occurrences that
would be affected if we ran the command

Normal Mode

Substituting text



Substitute a character



Substitute a line

Replacing Text



Replace a character



Replace over characters, one by one.

Vim Marks

Marks are hidden positions that allow us to jump back to that specific location or line.

m{a-zA-Z}

Lowercase Marks are shared within a file.

Uppercase Marks are shared among all files.



Backtick

Jump to exact position of the mark.



Single quote

Jump to the beginning of line that holds mark.

*Example: Suppose we have this text
and we want to mark different places.*

```
1 DOCUMENTATION
2   Framework3 Official Documentation.
3 INSTALLING
4   To install your framework, you
5   can use yarn, npm, or download ...
6 CONFIGURING
7   Main cofiguration file is placed at
8   local folder. There are two main ...
9 QUESTIONS
10  If you have any question or want to
11  talk about new features, send email ...
```

Setting up marks

```
1 md DOCUMENTATION
2   Framework3 Official Documentation.
3 mi INSTALLING
4   To install your framework, you
5   can use yarn, npm, or download ...
6 mc CONFIGURING
7   Main cofiguration file is placed at
8   local folder. There are two main ...
9 mq QUESTIONS
10  If you have any question or want to
11  talk about new features, send email ...
```

Navigating to the exact postion of mark.

```
1 `d DOCUMENTATION
2   Framework3 Official Documentation.
3
4 INSTALLING
5   To install your framework, you
   can use yarn, npm, or download .....
```

Navigation to the line that hold defined mark.

```
1
2
3 'i DOCUMENTATION
4   Framework3 Official Documentation.
5 INSTALLING
   To install your framework, you
   can use yarn, npm, or download .....
```

Navigation in Vim

Line Navigation

0

Beginning of the line

^

First non blank character

gg

First line on the file

\$

End of the line

g_

Last non blank character

G

Last line on the file

g;

Go to last edit

25G

Go to line 25

:50

Go to line 50

Line Scroll

Ctrl e

Scroll one line down

Ctrl y

Scroll one line up

Viewport Navigation

H

Head of the viewport

M

Middle of the viewport

L

Botom of the viewport

Tip: Press **zz** to position your highlight in the middle of viewport

Viewport Scroll

Ctrl d

Scroll half page down

Ctrl u

Scroll half page up

Ctrl f

Scroll one page down

Ctrl b

Scroll one page up

Mark Navigation

`m

Exact position of mark

'm

Go to the line that holds the mark

:marks

Lists all marks

Word Navigation

w

Beginning of the next word

e

To the end of next word

b

Beginning of the previous word

ge

To the end of previous word

Uppercase Version

Also, **W,E,B,gE** works in similar way, except that only non blank character are ignored. Try it!

Delete, Yank and Put In Vim

a.k.a Cut, Copy and Paste

Vim doesn't put text on clipboard, but in registers.

Normal Mode

X

Delete a character

dd

Delete a line

daw

Delete current word
including spaces

df_x

Delete including first
occurrence of char x

yy

Yank Line

y\$

Yank to the
end of line

yiw

Yank current word
without space

yt_x

Yank before first
occurrence of char x

p

Put After

P

Put Before

gp

Put After and move
cursor after pasted text

Command Mode

:%y

Yank all lines.

:1,20 co.

*Yank from line 1 to 20
and put it to cursor position*

Visual Mode

v

Highlight

V

Highlight a Line

Ctrl

V

Highlight a Block

d

Delete

y

Yank

p

Put

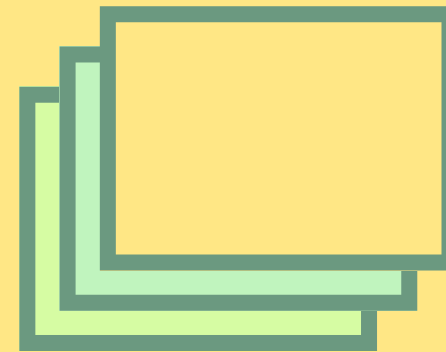
Vim Windows

A window is a viewport onto a file (buffer).

You can use multiple windows on one buffer, or several windows on different buffers



Multiple windows on one buffer



Several windows on different buffers

Command

:sp

Split Window

:vsp

Split Window Vertically

:sp *fileName*

Open a file in new window.

We can open multiple buffers in their respective windows:

```
vim -o App.js index.html main.css
```

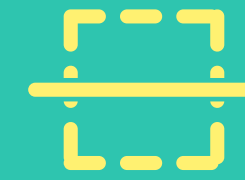
-O Uppercase O to open windows vertically.

Useful Keyboard shortcuts



Ctrl+W V

Split window vertically



Ctrl+W S

Split window horizontally



Ctrl+W W

Move between opened windows



Ctrl+W j

Move one window down



Ctrl+W k

Move one window up



Ctrl+W h

Move one window to the left



Ctrl+W l

Move one window to the right



Ctrl+W =

Resize windows to be of equal size



Ctrl+W q

Close active window

Vim Text Objects

Text objects allow us to interact with plain text, parentheses, quotes, tags, and other common patterns that appear in text.

Optional:
Number of time to repeat

<number><command><text object or motion>

Vim Command:
Change, Delete, Yank, Paste

Text Object or Motion:
can be a text, word, paragraph
or a motion.

Code	Keystrokes	Result
<div>const sqr = (a,b) => {}</div> <div>Cursor position</div>	i)	<div>Change inside a pair of parentheses</div> <div>const sqr = () => {}</div>
	a)	<div>Delete a pair of parentheses</div> <div>const sqr = => {}</div>
<div>let profile = {</div> <div>Cursor position</div> <div>name:"vim",</div> <div>version:7.5</div> <div>}</div>	i}	<div>Change inside a pair of curly braces.</div> <div>let profile = {</div>
	a}	<div>Delete a pair of curly braces.</div> <div>let profile = </div>
<div>var score = ["B","D","C"]</div> <div>Cursor position</div>	i]	<div>Change inside a pair of brackets.</div> <div>var score = []</div>
	a]	<div>Delete a pair of brackets.</div> <div>var score = </div>
<div>var name = "Vimnator";</div> <div>Cursor position</div>	i"	<div>Change inside a pair of quotes.</div> <div>var name = " "</div>
	a"	<div>Delete a pair of quotes.</div> <div>var name = </div>
<div><h2>Practical Vim</h2></div> <div>Cursor position</div>	it	<div>Change inside a pair of tags.</div> <div><h2></h2></div>
	at	<div>Delete a pair of tags.</div> <div></div>
<div>A word or a WORD?</div> <div>Cursor position</div>	iw	<div>Change/Delete/Yank a word.</div> <div>A or a WORD?</div>
	aw	<div>Change/Delete/Yank a word, including space around that word</div> <div>A or a WORD?</div>