vim

• • •

edit at the speed of thought

Why vim?

- > amazingly powerful
- > can be way faster than any other editor
- > installed everywhere

Why, vim?!



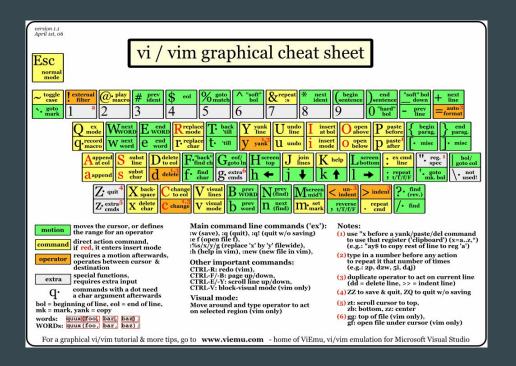
Always enjoy seeing someone trying to exit Vim for the first time.



AAAAAAAAAAAAHHHHHRHRGRGRGRRRGURBHJB EORWPSOJWPJORGWOIRGWSGODEWPGOHEPW09GJEDPOKSD!!!!!!!!!!!0924Q U8T63095JRGHWPE09UJ0PWHRGW

11:37 AM · Sep 18, 2018 · Twitter Web Client

Why, vim?!



Why, vim?!

- > what did I just press? What's happening?
- > why bother, when GUI editors are easier and faster?

It's easier than it seems

- > it has a logical grammar, like a language
- > a few simple elements that compose into powerful, flexible commands
- > there are a lot of more obscure commands too, but you can do a lot without them
- > don't try to memorize everything, just practice

Vim basics

modes

Main modes

- NORMAL MODE [esc]
 Use vim commands
- > INSERT MODE [i]
 Edit the text directly

Other modes

- > VISUAL MODE [v]
 Select blocks of text visually
- > REPLACE MODE [R]
 Like insert mode, but deletes characters as it goes
- > And more, but don't worry about those

Important shortcuts

```
u undo
Ctrl+r redo
Ctrl+g display a little info
```

Get set up

- > open (or make) a text file
- > type :set number to show line numbers
- > type :set showcmd to show the commands you type in the bottom right corner

Grammar summary

- > I'll go through these components one at a time
- > There's also colon commands, e.g. :q and :s/foo/bar. I'll come to these later.

motion

Move the cursor

Motions

More motions

Repeating motions

```
Prefixing a motion with a number repeats it

10w move to the next word 10 times

3↓ move 3 lines down

2/foo moves the the next occurence of foo twice
```

motion exercises

```
Try to go:
> forward 10 words
> back 5 words
> down 3 lines
> the end of a block of [square brackets]
> to the end of the last line
> to the 5th occurrence of foo in the file
```

action

actions

```
<u>c</u>hange (delete + insert)
```

Self-contained actions

```
D/C applies D/C to the end of the line

x deletes the character under the cursor
```

repetition

. repeats the last action

```
Just like with motions, adding an number before it repeats it, e.g. 10dd 10dw (or d10w)
```

You can repeat both actions and motions in the same command, e.g 10d10w

If the last action you did was inserting text, then . will repeat that, and 10. Will repeat it 10 times

action exercises

MOTIONS		ACTIONS	
	next <u>w</u> ord	d	<u>d</u> elete
b	<u>b</u> ack		<u>c</u> hange
	<pre>end of the current word</pre>		yank
	end of line	р	<u>p</u> aste
	start of line		indent
	the matching {[(dd/cc/yy	d/c/y whole line
	<u>g</u> o to line 10		d/c to the end line
gg	<u>g</u> o to start of file		delete character
	<u>g</u> o to end of file		replace character
	<u>f</u> ind x		
	<u>f</u> ind foo	Prefix with number to repeat . to repeat last action/insert	

Prefix with number to repeat

Try to:

- > delete the next 10 lines
- > replace all the text until the next 'a' with "foo"
- > copy and paste the current word
- > make 1000 lines of "all work and no play makes jack a dull boy"
- > use visual mode to select and delete a section of a word

Bonus: motion modifiers

Motion modifiers

```
i inside
a around (inside + end chars too)

Examples
    diw delete inside the word (i.e. the whole word)
    ci" change the text inside quotes
    ca{ the {} block, including the enclosing {}
```

other commands

Entering insert mode

```
a append (insert after cursor)

A Append to line
I Insert at the start of the line

o open a new line under the current
Open a new line above the current
```

Recording macros

Want to repeat movements + actions combined? Or a complex set of actions Make a macro

- 1. Press q to start recording
- 2. Press the key you want to map your macro to
- Make your change
- 4. Press q to stop recording
- 5. Use @[your char] to repeat your macros

Using :commands

Customizing vim

```
Like a setting? Store it in your vimrc. e.g.

Set number

syntax on

Set hlsearch
```

Plugins

Remapping commands

Summary

- > Vim is like a language with a small vocabulary and a simply grammar
- > It's not hard to learn, but it takes time and practice
- > It's more about combining simple elements than learning tonnes of complex ones
- > It's slower than what you're used to at first (but gets faster)
- > There's some irregular parts and things that don't quite do what they should. Experiment and learn

Next steps

- > learn the four main actions: c, d, y, p
- > learn a few motions, or keep a cheat sheet handy
- > practice using repetition when possible
- > gradually expand your repertoire
- > vimtutor