

Shortcuts to move faster in Bash command line

Jan 4, 2012

Nowadays, I spend more time in Bash shell, typing longer commands. One of my new year resolutions for this year is to **stop using left/right arrow keys to move around in the command line**. I learned [a few shortcuts](#) a while ago.

Last night, I spent some time to read about “**Command Line Editing**” in [the bash manual](#). The bash manual is a well-written piece of documentation. I think I should read it more often.

Well, here's the new shortcuts I learned:

Basic moves

- Move back one character. `Ctrl + b`
- Move forward one character. `Ctrl + f`
- Delete current character. `Ctrl + d`
- Delete previous character. `Backspace`
- Undo. `Ctrl + -`

Moving faster

- Move to the start of line. `Ctrl + a`
- Move to the end of line. `Ctrl + e`
- Move forward a word. `Meta + f` (*a word contains alphabets and digits, no symbols*)
- Move backward a word. `Meta + b`
- Clear the screen. `Ctrl + l`

What is Meta? `Meta` is your `Alt` key, normally. For **Mac OSX user**, you need to enable it yourself. Open *Terminal > Preferences > Settings > Keyboard*, and enable *Use option as meta key*. `Meta` key, by convention, is used for operations on word.

Cut and paste (‘Kill and yank’ for old schoolers)

- Cut from cursor to the end of line. `Ctrl + k`
- Cut from cursor to the end of word. `Meta + d`
- Cut from cursor to the start of word. `Meta + Backspace`
- Cut from cursor to previous whitespace. `Ctrl + w`
- Paste the last cut text. `Ctrl + y`
- Loop through and paste previously cut text. `Meta + y` (use it after `Ctrl + y`)
- Loop through and paste the last argument of previous commands. `Meta + .`

Search the command history

- Search as you type. `Ctrl + r` and type the search term; Repeat `Ctrl + r` to loop through results.
- Search the last remembered search term. `Ctrl + r` twice.
- End the search at current history entry. `Ctrl + j`
- Cancel the search and restore original line. `Ctrl + g`

Need more?

- A comprehensive [bash editing mode cheatsheet](#) by Peteris Krumin ([catonmat.net](#)).
- **Vim users!** Do you know you can switch to Vi-style editing mode? Here: [vi-style cheatsheet](#).
- Bash command line editing is actually handled by **GNU Readline Library**. So just dive into [Readline manual](#) for everything else.

33 Comments

teohm.com/blog

 Login ▾ Recommend 6 Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name

**Boldewyn** • 5 years ago

If you're a Vim user, you might be interested in the part "Readline vi mode",
<http://www.gnu.org/software...>

9 ^ | ▾ • Reply • Share ›

**Huiming Teo** author ➔ **Boldewyn** • 5 years ago

I just turned on the vi mode for bash editing line, going to try out for a week. :)

1 ^ | ▾ • Reply • Share ›



Boldewyn → Huiming Teo • 5 years ago

A word of warning: The vi editing mode is a bit neglected compared to the default Emacs mode. There are rough edges and it doesn't respond as smoothly as you expect from Vim. But I enjoy the consistency all the more.

^ | v • Reply • Share ›



Kenneth Younger III → Boldewyn • 5 years ago

If you're serious about using vim mode at a command line, zsh would be much more worth your time.

^ | v • Reply • Share ›



Huiming Teo author → Kenneth Younger III • 5 years ago

Kenneth, why do you say so?

^ | v • Reply • Share ›



Guillaume • 5 years ago

These are actually readline shortcuts, not solely bash ones. That means they should work in any applications that uses readline (gdb or the mysql command line client for example)

6 ^ | v • Reply • Share ›



Bosco mutunga → Guillaume • 5 years ago

emacs too, similar keymap

1 ^ | v • Reply • Share ›



Huiming Teo author → Guillaume • 5 years ago

Thanks for the tips!

^ | v • Reply • Share ›



bashnoob → Guillaume • 5 years ago

Ahh that would be why Emacs uses the same shortcuts...

^ | v • Reply • Share ›



jrdmcgr • 5 years ago

esc will also work in place of the meta key, but you must press and release esc then press f or b.

2 ^ | v • Reply • Share ›



jrdmcgr → jrdmcgr • 5 years ago

Also, ctrl + u will cut to the beginning of the line.

1 ^ | v • Reply • Share ›



Josh • 5 years ago

Not exactly a movement trick, but useful when you make mistakes, want to run many similar commands in sequence, or edit blocks of shell: fc for Fix Command. Brings up your last command in your EDITOR of choice.

1 ^ | v • Reply • Share ›



Huiming Teo author → Josh • 5 years ago

Thanks Josh, that's a useful one!

^ | v • Reply • Share ›



inquist • 5 years ago

set -o vi

1 ^ | v • Reply • Share ›



djaka pribadi maulana • 6 years ago

Are those shortcuts valid for any OS?

1 ^ | v • Reply • Share ›



Adam Covati → djaka pribadi maulana • 5 years ago

These shortcuts are pretty much an essential part of my day. I cringe when I see people press and hold the arrow key for 20 seconds to go to the beginning of the line.

And yes, many of these actually work in most text entry fields on OSX. For example, I just used ctrl-e, ctrl-a, and ctrl-k within this discussion field.

2 ^ | v • Reply • Share ›



tmt → Adam Covati • 4 years ago

worst part? Ctrl+w is delete last word, I use it all the time on the shell

I've lost count on how many times I closed the tab in which I was editing a text-field because of this shortcut....

1 ^ | v • Reply • Share ›



Samuel → tmt • 2 months ago

@tmt This also happens to me! I keep killing the IPython tab when I'm using Spyder.

^ | v • Reply • Share ›



Huiming Teo author → djaka pribadi maulana • 6 years ago

Ya, it should work in Bash terminal across OSs such as Linux, Mac OSX, not sure about Windows though. :)

^ | v • Reply • Share ›



ano → Huiming Teo • 5 years ago

Some terminal emulators have menu shortcut keys with them, so they have to be disabled to be able to use shortcuts like Meta+f.

^ | v • Reply • Share ›



Frostbeard → Huiming Teo • 5 years ago

They definitely work in Cygwin's bash on Win7 at least.

^ | v • Reply • Share ›



Jesse Pinkman → Frostbeard • 4 years ago



Walter??

1 ^ | v • Reply • Share ›

**Sten** • a month ago

Nice! Thanks!

^ | v • Reply • Share ›

**Abel Pc Ag** • 4 months ago

If you're an iTerm2 user, you maybe want to use `_ ←` and `_ →` to jump forwards / backwards words.

<https://coderwall.com/p/h6y...>

^ | v • Reply • Share ›

**Huge** • 7 months ago

Ctrl+R to grab what I currently wrote on the line and trying to finish?

^ | v • Reply • Share ›

**Edgar KAMDEM** • a year ago

Please how to enable meta key on IntelliJ IDEA embedded terminal ?

^ | v • Reply • Share ›

**lingtalfi** • 2 years ago

For mac users, be more productive and use my shorcuts here:

<https://github.com/lingtalf...>

ALT-left: move one word backward

ALT-right: move one word forward

CTRL-left: move to the beginning of the line

CTRL-right: move to the end of the line

ALT-backspace: kill one word backward

ALT-del: kill one word forward

ALT-up: set word after cursor to uppercase

ALT-down: set word after cursor to lowercase

home: move to the beginning of the line

end: move to the end of the line

CTRL-backspace: Same as ALT-backspace

CTRL-del: Same as ALT-del

^ | v • Reply • Share ›

**vincent** • 3 years ago

Good~and it sounds interesting that turn on the vi mode for bash editing.

^ | v • Reply • Share ›

**nerglish** • 3 years ago

for mac/iterm2 users, set Preferences > Profiles > Keys > Left/Right option key acts as "+Esc"

^ | v • Reply • Share ›

**Terex** • 4 years ago

cool

^ | v • Reply • Share ›

**hellekin** • 5 years ago

I use C-t and M-t often to correct typos. Also, M-Left and M-Right moves to previous and next word respectively.

^ | v • Reply • Share ›

**Phillip** • 5 years ago

Ctrl + j, Ctrl + m both work as RETURN keys. Saves your right pinky a bit once you get used to it. You can also perform Emacs style incremental search with Ctrl + s on the current line.

^ | v • Reply • Share ›

**Dmitry Cheryasov** • 5 years ago

End the search at current history entry: just press Enter :)

Also important: when you've searched all the way up the history and want to get back to its very end, press Meta + >.

^ | v • Reply • Share ›

ALSO ON TEOHM.COM/BLOG

Start Using Ruby % (Percent) Notation

4 comments • 5 years ago

Avatar **Joe Fredette** — I thought you were going to talk about the printf-style string subbing, which is also a very cool trick. "%s: %d" %

Ruby idiom to ensure variable is Array - teohm.dev

13 comments • 5 years ago

Breaking ARC retain cycle in Objective-C blocks - teohm.dev

5 comments • 5 years ago

Avatar **Andrew Carter** — Another neat thing you can do is to __weak __typeof__(self) selfinside of the block, which will make any

Chef cookbooks for busy Ruby developers - teohm.dev

29 comments • 4 years ago

Hi, I'm Huiming!

Contact me via GitHub or

Twitter.



I build web apps. Currently a full-time dev dad.

This is my personal site for experimental work.

I write once a while, more often I tweet good reads about tech & life.