

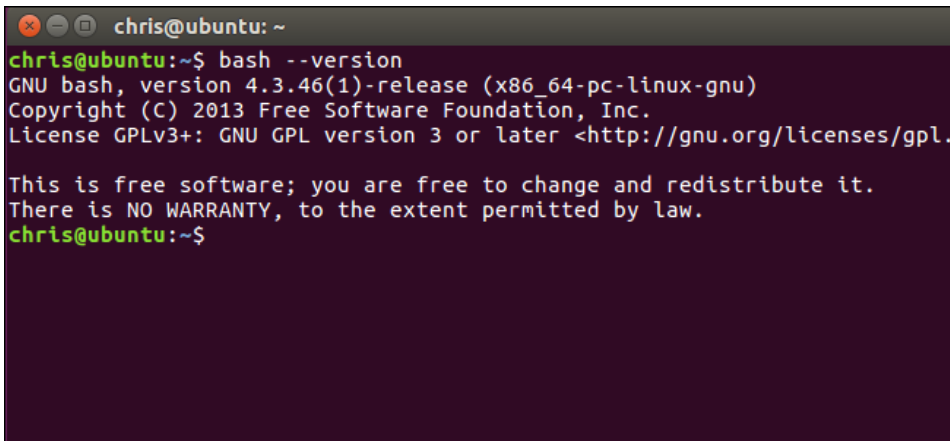
How-To Geek

The Best Keyboard Shortcuts for Bash (aka the Linux and macOS Terminal)



LOWELL HEDDINGS [@lowellheddings](#)

MARCH 17, 2017, 6:40AM EDT



```
chris@ubuntu: ~  
chris@ubuntu:~$ bash --version  
GNU bash, version 4.3.46(1)-release (x86_64-pc-linux-gnu)  
Copyright (C) 2013 Free Software Foundation, Inc.  
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.  
This is free software; you are free to change and redistribute it.  
There is NO WARRANTY, to the extent permitted by law.  
chris@ubuntu:~$
```

Bash is the default [command-line shell](#) on most Linux distributions, from Ubuntu and Debian to Red Hat and Fedora. Bash is also the default shell included with macOS, and you can [install a Linux-based bash environment on Windows 10](#).

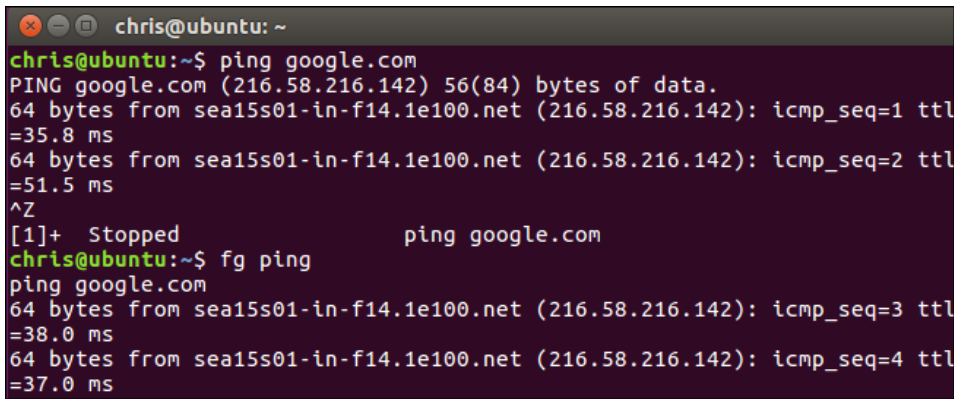
The bash shell features a wide variety of keyboard shortcuts you can use. These will work in bash on any operating system. Some of them may not work if you're accessing bash remotely through an SSH or telnet session, depending on how you have your keys mapped.

Working With Processes

Use the following shortcuts to manage running processes.

- **Ctrl+C:** Interrupt (kill) the current foreground process running in the terminal. This sends the SIGINT signal to the process, which is technically just a request—most processes will honor it, but some may ignore it.

- **Ctrl+Z**: Suspend the current foreground process running in bash. This sends the SIGTSTP signal to the process. To return the process to the foreground later, use the `fg process_name` command.
- **Ctrl+D**: Close the bash shell. This sends an EOF (End-of-file) marker to bash, and bash exits when it receives this marker. This is similar to running the `exit` command.

A terminal window titled 'chris@ubuntu: ~' with a dark background. The user runs 'ping google.com'. The output shows two successful pings. Then, the user presses Ctrl+Z, which sends the process to the background. The prompt changes to '[1]+ Stopped ping google.com'. The user then presses 'fg', which brings the ping process back to the foreground. The prompt returns to 'chris@ubuntu:~\$' and the ping command resumes, showing two more successful pings.

```
chris@ubuntu:~$ ping google.com
PING google.com (216.58.216.142) 56(84) bytes of data.
64 bytes from sea15s01-in-f14.1e100.net (216.58.216.142): icmp_seq=1 ttl
=35.8 ms
64 bytes from sea15s01-in-f14.1e100.net (216.58.216.142): icmp_seq=2 ttl
=51.5 ms
^Z
[1]+  Stopped                  ping google.com
chris@ubuntu:~$ fg ping
ping google.com
64 bytes from sea15s01-in-f14.1e100.net (216.58.216.142): icmp_seq=3 ttl
=38.0 ms
64 bytes from sea15s01-in-f14.1e100.net (216.58.216.142): icmp_seq=4 ttl
=37.0 ms
```

Controlling the Screen

The following shortcuts allow you to control what appears on the screen.

- **Ctrl+L**: Clear the screen. This is similar to running the “clear” command.
- **Ctrl+S**: Stop all output to the screen. This is particularly useful when running commands with a lot of long, verbose output, but you don’t want to stop the command itself with Ctrl+C.
- **Ctrl+Q**: Resume output to the screen after stopping it with Ctrl+S.

Moving the Cursor

Use the following shortcuts to quickly move the cursor around the current line while typing a command.

- **Ctrl+A** or **Home**: Go to the beginning of the line.
- **Ctrl+E** or **End**: Go to the end of the line.
- **Alt+B**: Go left (back) one word.
- **Ctrl+B**: Go left (back) one character.

- **Alt+F**: Go right (forward) one word.
- **Ctrl+F**: Go right (forward) one character.
- **Ctrl+XX**: Move between the beginning of the line and the current position of the cursor. This allows you to press Ctrl+XX to return to the start of the line, change something, and then press Ctrl+XX to go back to your original cursor position. To use this shortcut, hold the Ctrl key and tap the X key twice.

Deleting Text

Use the following shortcuts to quickly delete characters:

- **Ctrl+D** or **Delete**: Delete the character under the cursor.
- **Alt+D**: Delete all characters after the cursor on the current line.
- **Ctrl+H** or **Backspace**: Delete the character before the cursor.

Fixing Typos

These shortcuts allow you to fix typos and undo your key presses.

- **Alt+T**: Swap the current word with the previous word.
- **Ctrl+T**: Swap the last two characters before the cursor with each other. You can use this to quickly fix typos when you type two characters in the wrong order.
- **Ctrl+_**: Undo your last key press. You can repeat this to undo multiple times.

Cutting and Pasting

Bash includes some basic cut-and-paste features.

- **Ctrl+W**: Cut the word before the cursor, adding it to the clipboard.
- **Ctrl+K**: Cut the part of the line after the cursor, adding it to the clipboard.
- **Ctrl+U**: Cut the part of the line before the cursor, adding it to the clipboard.

- **Ctrl+Y:** Paste the last thing you cut from the clipboard. The y here stands for “yank”.

Capitalizing Characters

The bash shell can quickly convert characters to upper or lower case:

- **Alt+U:** Capitalize every character from the cursor to the end of the current word, converting the characters to upper case.
- **Alt+L:** Uncapitalize every character from the cursor to the end of the current word, converting the characters to lower case.
- **Alt+C:** Capitalize the character under the cursor. Your cursor will move to the end of the current word.

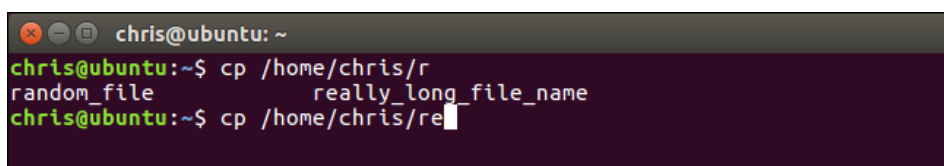
Tab Completion

RELATED: [*Use Tab Completion to Type Commands Faster on Any Operating System*](#)

[Tab completion](#) is a very useful bash feature. While typing a file, directory, or command name, press Tab and bash will automatically complete what you’re typing, if possible. If not, bash will show you various possible matches and you can continue typing and pressing Tab to finish typing.

- **Tab:** Automatically complete the file, directory, or command you’re typing.

For example, if you have a file named `really_long_file_name` in `/home/chris/` and it’s the only file name starting with “r” in that directory, you can type `/home/chris/r`, press Tab, and bash will automatically fill in `/home/chris/really_long_file_name` for you. If you have multiple files or directories starting with “r”, bash will inform you of your possibilities. You can start typing one of them and press “Tab” to continue.

A terminal window with a dark background. The prompt is 'chris@ubuntu: ~'. The user has typed 'cp /home/chris/r' and pressed Tab, which has completed the path to 'cp /home/chris/really_long_file_name'. The user then typed 'random_file' and pressed Tab, which also completed the path to 'cp /home/chris/really_long_file_name'.

```
chris@ubuntu: ~  
chris@ubuntu:~$ cp /home/chris/r  
random_file      really_long_file_name  
chris@ubuntu:~$ cp /home/chris/re
```

Working With Your Command History

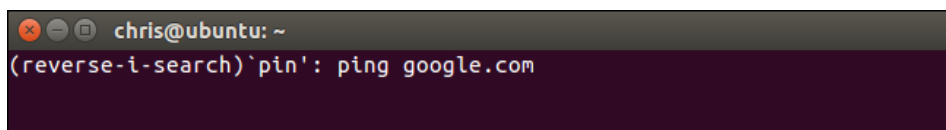
RELATED: [How to Use Your Bash History in the Linux or macOS Terminal](#)

You can quickly scroll through your recent commands, which are stored in your user account's [bash history](#) file:

- **Ctrl+P** or **Up Arrow**: Go to the previous command in the command history. Press the shortcut multiple times to walk back through the history.
- **Ctrl+N** or **Down Arrow**: Go to the next command in the command history. Press the shortcut multiple times to walk forward through the history.
- **Alt+R**: Revert any changes to a command you've pulled from your history if you've edited it.

Bash also has a special "recall" mode you can use to search for commands you've previously run:

- **Ctrl+R**: Recall the last command matching the characters you provide. Press this shortcut and start typing to search your bash history for a command.
- **Ctrl+O**: Run a command you found with Ctrl+R.
- **Ctrl+G**: Leave history searching mode without running a command.



emacs vs. vi Keyboard Shortcuts

The above instructions assume you're using the default keyboard shortcut configuration in bash. By default, bash uses emacs-style keys. If you're more used to the vi text editor, you can switch to [vi-style keyboard shortcuts](#).

The following command will put bash into vi mode:

```
set -o vi
```

The following command will put bash back into the default emacs mode:

```
set -o emacs
```

```
chris@ubuntu: ~  
chris@ubuntu:~$ set -o vi  
chris@ubuntu:~$ set -o emacs  
chris@ubuntu:~$
```

With a few of these in your toolbelt, you'll be a Terminal master in no time.

READ NEXT

- › [How to Use the fold Command on Linux](#)
- › [How to Hide Private Photos and Videos on Your iPhone or iPad](#)
- › [How to Stream the 92nd Annual Academy Awards](#)
- › [What Does "FTFY" Mean, and How Do You Use It?](#)
- › [Web Browsers Are Silencing Annoying Notification Popups](#)



LOWELL HEDDINGS

Lowell is the founder and CEO of How-To Geek. He's been running the show since creating the site back in 2006. Over the last decade, Lowell has personally written more than 1000 articles

which have been viewed by over 250 million people. Prior to starting How-To Geek, Lowell spent 15 years working in IT doing consulting, cybersecurity, database management, and programming work. [READ FULL BIO »](#)

How-To Geek is where you turn when you want experts to explain technology. Since we launched in 2006, our articles have been read more than 1 billion times. [Want to know more?](#)

