

[bash](#), [howto](#), [commandlineediting](#), [keyboard](#), [shortcut](#)



Bash - Command Line Editing

Sources

- <https://ss64.com/bash/syntax-keyboard.html>
- <https://www.howtogeek.com/howto/ubuntu/keyboard-shortcuts-for-bash-command-shell-for-ubuntu-debian-suse-redhat-linux-etc/>

Basic Shortcuts

You can use the following key combinations to edit and recall commands:

Shortcut	Command
CTRL + L	Clear the screen.
CTRL + W	Delete the word starting at cursor.
CTRL + U	Clear the line i.e. Delete the all words from command line.
↑, ↓	Recall commands (see command history).
TAB	Auto-complete files, directory, command names and much more.
CTRL + R	Search through previously used commands (see command history)
CTRL + C	Cancel currently running commands.
CTRL + T	Swap the last two characters before the cursor.
ESC + T	Swap the last two words before the cursor.

Bash Keyboard Shortcuts

Moving the cursor

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

Shortcut	Command
Ctrl + A, POS1	Go to the beginning of the line (Home)
Ctrl + E, END	Go to the End of the line (End)
ALT + B	Back (left) one word
ALT + F	Forward (right) one word
Ctrl + F	Forward one character
Ctrl + B	Backward one character
Ctrl + Xx	(double tap 'X') 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 and tap X twice.

Editing

Shortcut	Command
CTRL + L	Clear the Screen, similar to the clear command
ALT + BACKSPACE	Delete the Word before the cursor.
CTRL + D	Delete character under the cursor
ALT + D	Delete from actual cursor position to the end of word
CTRL + H	Delete character before the cursor (Backspace)
CTRL + W	Cut the Word before the cursor to the clipboard.
CTRL + K	Cut the Line after the cursor to the clipboard.
CTRL + U	Cut/delete the Line before the cursor to the clipboard.
ALT + T	Swap current word with previous
CTRL + T	Swap the last two characters before the cursor (typo).
ESC + T	Swap the last two words before the cursor.
CTRL + Y	Paste the last thing to be cut (yank)
ALT + U	UPPER capitalize every character from the cursor to the end of the current word.

Shortcut	Command
ALT + L	Lower the case of every character from the cursor to the end of the current word.
ALT + C	Capitalize the character under the cursor and move to the end of the word.
ALT + R	Cancel the changes and put back the line as it was in the history (revert).
CTRL +]	Undo
TAB	Tab completion for file/directory names For example, to move to a directory 'sample1'; Type cd sam ; then press TAB and ENTER. type just enough characters to uniquely identify the directory you wish to open.

Special keys: Tab, Backspace, Enter, Esc

Text Terminals send characters (bytes), not key strokes. Special keys such as **Tab**, **Backspace**, **Enter** and **Esc** are encoded as control characters. Control characters are not printable, they display in the terminal as ^ and are intended to have an effect on applications.

Keys	Control Character
Ctrl + I	TAB
Ctrl + J	Newline
Ctrl + M	ENTER
Ctrl + [ESC

Many terminals will also send control characters for keys in the digit row:

Shortcut	Result
CTRL + 2	^@
CTRL + 3	^[ESCAPE
CTRL + 4	^\
CTRL + 5	^]
CTRL + 6	^^
CTRL + 7	^_ UNDO
CTRL + 8	^? Backward-delete-char

CTRL+**V** tells the terminal to not interpret the following character, so **CTRL**+**V** **CTRL**+**I** will display a **TAB** character, similarly **CTRL**+**V** **ENTER** will display the escape sequence for the Enter key: **^M**.

History

Shortcut	Command
CTRL + R or ↑	Recall the last command including the specified character(s). Searches the command history as you type. Equivalent to : vim ~/.bash_history.
CTRL + P or ↓	Previous command in history (i.e. walk back through the command history)
CTRL + N	Next command in history (i.e. walk forward through the command history)
CTRL + Shift	Go back to the next most recent command. (beware to not execute it from a terminal because this will also launch its XOFF).
CTRL + O	Execute the command found via Ctrl + R or Ctrl + S
CTRL + G	Escape from history searching mode
!!	Repeat last command
!abc	Run last command starting with abc
!abc:p	Print last command starting with abc
!\$	Last argument of previous command
ALT + .	Last argument of previous command
!*	All arguments of previous command
^abc^def	Run previous command, replacing abc with def

Process control

Shortcut	Command
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 + 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.
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.

Controlling the screen

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

Shortcut	Command
<code>CTRL+L</code>	Clear the screen. This is similar to running the “clear” command.
<code>CTRL+S</code>	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
<code>CTRL+Q</code>	Resume output to the screen after stopping it with <code>Ctrl+Shift</code> .

Emacs mode vs Vi Mode

All the above assume that bash is running in the **default Emacs setting**, if you prefer this can be switched to **Vi shortcuts** instead.

Set **Vi Mode** in bash:

```
set -o vi
```

Set **Emacs Mode** in bash:

```
set -o emacs
```

“...emacs, which might be thought of as a thermonuclear word processor” ~ [Emacs vs. Vi Wiki](#)

Related

- [fg](#) - Bring a command to the foreground.
- [vi editor](#) - [A one page reference to the vi editor](#).
- `~/.bash_history` - Text file with command history.
- [Terminals Are Weird](#) - How and why of terminal keybindings.
- Equivalent [Windows Keyboard shortcuts](#)

From:
<https://wiki.nanoscopic.de/> - **nanoscopic wiki**

Permanent link:
<https://wiki.nanoscopic.de/doku.php/pages/howtos/bash/bash-command-line-editing?rev=1613047591>

Last update: **2021/02/11 12:46**

