

Bash Keyboard Shortcuts

Moving the cursor:

```
Ctrl + a    Go to the beginning of the line (Home)
Ctrl + e    Go to the End of the line (End)
Ctrl + p    Previous command (Up arrow)
Ctrl + n    Next command (Down arrow)
Alt + b     Back (left) one word
Alt + f     Forward (right) one word
Ctrl + f    Forward one character
Ctrl + b    Backward one character
Ctrl + xx   Toggle between the start of line and current cursor position
```

Editing:

```
Ctrl + L    Clear the Screen, similar to the clear command

Alt + Del   Delete the Word before the cursor.
Alt + d     Delete the Word after the cursor.
Ctrl + d    Delete character under the cursor
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.
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.

History:

```
Ctrl + r    Recall the last command including the specified character(s)
             searches the command history as you type.
             Equivalent to : vim ~/.bash_history.

Ctrl + p    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 + s    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:

```
Ctrl + C    Interrupt/Kill whatever you are running (SIGINT)
Ctrl + l    Clear the screen
Ctrl + s    Stop output to the screen (for long running verbose commands)
             Then use PgUp/PgDn for navigation
Ctrl + q    Allow output to the screen (if previously stopped using command above)
Ctrl + D    Send an EOF marker, unless disabled by an option, this will close the current shell (EXIT)
Ctrl + Z    Send the signal SIGTSTP to the current task, which suspends it.
             To return to it later enter fg 'process name' (foreground).
```

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](#)

