Bash Shortcuts

These are Readline (Emacs-mode) key bindings — they work the same in Bash, Zsh, and most REPLs, on Linux and macOS. Learning even the top row (`Ctrl-A` , `Ctrl-R` , `Ctrl-W`) makes the shell dramatically faster.

Move the Cursor

Keys	Moves to
`Ctrl-A`	start of the line
`Ctrl-E`	end of the line
`Alt-B`	back one word
`Alt-F`	forward one word
`Ctrl-XX`	toggle between start and current position

Edit the Line

Keys	Does
`Ctrl-W`	delete the word before the cursor
`Ctrl-U`	delete from cursor to start of line
`Ctrl-K`	delete from cursor to end of line
`Ctrl-Y`	paste (yank) what you just deleted
`Alt-D`	delete the word after the cursor
`Ctrl-T`	transpose (swap) the last two characters
`Alt-U / Alt-L`	uppercase / lowercase the next word
`Ctrl-_`	undo the last edit

> **Cut and paste, shell-style.** `Ctrl-U` , `Ctrl-K` , and `Ctrl-W` don't just delete — they cut into a buffer. `Ctrl-Y` pastes it back, so you can move text around the line.

History

Keys / token	Does
`Ctrl-R`	reverse-search history (type to filter)
`Ctrl-G`	cancel the history search
`Ctrl-P / Ctrl-N`	previous / next command
`Alt-.`	insert the last argument of the previous command
`!!`	the entire previous command
`!$`	the last argument of the previous command
`!abc`	the most recent command starting with abc

The classic sudo rescue

```
apt update          # oops, permission denied
sudo !!             # re-runs: sudo apt update
```

Control the Shell

Keys	Does
`Ctrl-C`	interrupt the running command
`Ctrl-D`	end of input / log out of the shell
`Ctrl-Z`	suspend to the background (`fg` to resume)
`Ctrl-L`	clear the screen (same as `clear`)
`Ctrl-S / Ctrl-Q`	pause / resume terminal output