



The Ultimate 'Linux Commands' Cheatsheet

Welcome to the [Ultimate 'Linux Commands' Cheatsheet](#) from **Josh Bruton** @ bruchant.net

[A Brutopia Design & Development Project](#)

"The terminal is your superpower. Don't try to memorize it all — keep this cheatsheet close and let efficiency guide your journey."

What are Commands in Linux & Why Should You Care?

Learning the **Bash/Linux** command line is one of the most **powerful** steps you can take as a programmer. The command line strips away the distractions and puts efficiency front and center — letting you move **faster**, **automate repetitive tasks**, and **truly understand** what's happening under the hood of your computer.

For many developers (myself included), Linux was the doorway into programming itself. It's not about memorizing *every keyword* or *obscure flag* — that's **impossible**, and **unnecessary**. Instead, it's about building confidence with the tools, knowing how to search, and keeping resources like this cheatsheet at your side. Think of it as your compass: a quick reference that helps you navigate the terminal with clarity, speed, and purpose on your coding journey.

Bash Keywords

Bash (Bourne Again SHell) is both a *programming language* AND a *command-line interpreter* that's been around longer than Python. While Python is known for **general-purpose programming**, Bash is designed for **navigating** and **controlling** the Linux shell. **Both** can be run interactively — Python through its **REPL** and Bash directly in the **terminal** — or written as scripts to automate tasks. In Bash, scripts are made executable and run right inside the command line, making it a powerful tool for system control, automation, and efficiency.

I have broken the Bash Keywords down into Categories to make referencing much quicker.

ESSENTIALS

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>pwd</code>	Print Working Directory	Show the current directory at the command line
<code>cd <dir></code>	Change Directory	Go to <directory>
<code>ls</code>	List files	Shows all the files that exist in the current <dir>
<code>history</code>	Print the terminal's history	List your most recent commands in the terminal
<code>man <command></code>	Show Manual for a command	Output documentation in terminal for any <command> that has a manual

FILES & DIRECTORIES

These commands are used to browse the Linux file tree from a terminal. As well as create, delete, move, copy files/directories.

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>mkdir <dir></code>	Create new folder	Create new folder named <dir>
<code>touch <filename></code>	Create new empty File	Create new <file> named <filename>
<code>rm <file></code>	Remove File	Delete <file> from current working directory
<code>rm -r <dir></code>	Remove Folder	Delete <dir> recursively (all files/folders inside)
<code>mv <src> <dst></code>	Move File	Move <file> from <src> to < dst >
<code>cp <src> <dst></code>	Copy File	Copy <file> from < src > to < dst >

VIEWING FILES

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>cat <file></code>	Concatenate and print files	Print all the contents of <file> to standard output (the terminal)
<code>more <file></code>	Show with Pagination	Show chunks of a <file> at a time

<code>less <file></code>	View file, scrollable	Show chunks of a <code><file></code> at a time
<code>tail <file></code>	Displays end of a file	Prints the last byte of a <code><file></code>
<code>more <file></code>	Output's chunks at a time of large files	Displays the content of the <code><file></code> , page per page, from the beginning.
<code>head <file></code>	Output's 10 lines of the file starting at the beginning	Similar to <code>tail</code> but to display the beginning of the <code><file></code> .

SEARCH & TEXT PROCESSING

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>grep <str> <file></code>	Search text	Find lines (within <code><file></code>) that include <code><str></code>
<code>grep -r <str> <dir></code>	Recursive search	Search all files in <code><dir></code>
<code>find <dir> -name <file></code>	Recursive search	Search all files in <code><dir></code>
<code>locate <file></code>	Fast file search	Uses prebuilt index
<code>wc -l <file></code>	Line count	Count lines in file
<code>sort <file></code>	Sort lines	Outputs sorted content
<code>uniq <file></code>	Unique lines	Removes duplicates
<code>cut -d: -f1 <file></code>	Extract fields	Split file by delimiter

ARCHIVING & COMPRESSION

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>tar -cvf archive.tar files/</code>	Create archive	Bundle files into tar
<code>tar -xvf archive.tar</code>	Extract archive	Unpack tar contents
<code>tar -czvf archive.tar.gz files/</code>	Create compressed archive	Tar + gzip
<code>tar -czvf archive.tar.gz</code>	Extract compressed	Unpack <code>.tar.gz</code>
<code>zip archive.zip files</code>	Create Zip	Compress files into zip
<code>unzip archive.zip</code>	Extract Zip	Unpack zip archive

NETWORK COMMANDS

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>ifconfig</code>	Outputs your current network config	May need <code>net-tools</code>
<code>ip a</code>	Show interfaces	Modern alternative to ifconfig
<code>iwconfig</code>	Show Wi-Fi info	Wireless SSID, speed, etc.
<code>ping <host></code>	Check if host is alive	Check if host is reachable
<code>hostname</code>	Show/set hostname	Without args → show current
<code>wget <url></code>	The non-interactive network downloader	Works with APIs, web requests

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>netstat</code>	Show connections	Obsolete (use <code>ss</code>)
<code>ss</code>	Socket statistics	Replacement for <code>netstat</code>
<code>ip route</code>	Show routing	Replacement for <code>netstat -r</code>
<code>ssh <user>@<ip></code>	Remote login	OpenSSH remote shell
<code>scp <src> <user>@<host>:<dst></code>	Copy over SSH	Secure file transfer

The `netstat` program is mostly obsolete.

Replacement for `netstat` is `ss`. Replacement for `netstat -r` is `ip route`.

Replacement for `netstat -i` is `ip -s link`. Replacement for `netstat -g` is `ip addr`

PROCESS MANAGEMENT

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>ps aux</code>	Show processes	Snapshot of running processes
<code>top</code>	Process monitor	Live system view
<code>htop</code>	Enhanced top	Interactive (install separately)
<code>kill <pid></code>	Kill process	End process by ID
<code>killall <name></code>	Kill by name	End all the <code><name></code>
<code>jobs</code>	Background jobs	Show jobs in current shell
<code>fg %<n></code>	Foreground job	Resume job in foreground
<code>bg %<n></code>	Background job	Resume job in background

USERS & PERMISSIONS

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>whoami</code>	Current user	Show active username
<code>id</code>	User info	UID, GID, groups
<code>chmod 755 <file></code>	Change permissions	<code>rw-r-xr-x</code> example
<code>chown user:group <file></code>	Change ownership	Set file owner/group
<code>sudo <command></code>	Run as root	Execute with privileges
<code>passwd</code>	Change password	For current user
<code>adduser <name></code>	Add user	Create new account
<code>deluser <name></code>	Remove user	Delete account

SYSTEM INFORMATION

Here are the main commands to know, in order to manage and use the network on your computer or server.

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>uname -a</code>	Kernel/system info	Version, architecture
<code>hostnamectl</code>	Hostname info	Works on systemd distros
<code>df -h</code>	Disk usage	Human-readable free space
<code>du -sh <dir></code>	Folder size	Total size of directory
<code>free -h</code>	Memory usage	Free/used memory
<code>uptime</code>	System uptime	Time since boot
<code>dmesg</code>	Kernel logs	Boot/hardware messages
<code>journalctl -xe</code>	System logs	Detailed logs

PACKAGE MANAGEMENT (Debian/Ubuntu)

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>apt update</code>	Refresh repos	Update package lists
<code>apt upgrade</code>	Upgrade system	Update installed packages
<code>apt install <pkg></code>	Install software	Install new package
<code>apt remove <pkg></code>	Remove package	Uninstall package
<code>apt search <pkg></code>	Search packages	Find software in repos

REDIRECTION & PIPES

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>></code>	Redirect output	Overwrite files with output
<code>>></code>	Append output	Append output to file
<code><</code>	Redirect input	Use file as input
<code> </code>	Pipe	When finished with previous code move to next
<code>tee <file></code>	Write + view	Output to file and screen

SCRIPTING BASICS

SYNTAX	FUNCTIONALITY	DESCRIPTION
<code>echo "Hello"</code>	Print text	Output text to terminal
<code>#!/bin/bash</code>	Shebang	First line of script
<code>chmod +x script.sh</code>	Make executable	Run with <code>./script.sh</code>
<code>VAR=value</code>	Set variable	Assigns a variable
<code>\$VAR</code>	Use variable	Access variable value
<code>for i in {1..5}; do echo \$i; done</code>	Loop	Example: print 1-5
<code>if [condition]; then ... fi</code>	Conditional	Basic if-conditional structure

