

Terminal Mastery Guide – Partha's Personalized Handbook

This guide consolidates **all 11 terminal sections** you've mastered, along with key commands, explanations, and **practice questions** for each section.

Section 1: Basic Navigation & Files

Key Commands

Command	Description
<code>pwd</code>	Show current directory
<code>ls</code>	List directory contents
<code>ls -l</code>	Long listing with permissions
<code>ls -a</code>	Show hidden files
<code>cd dir/</code>	Change directory
<code>cd ..</code>	Go up one level
<code>mkdir folder</code>	Make new directory
<code>touch file.txt</code>	Create empty file
<code>rm file</code>	Remove file
<code>rmdir folder</code>	Remove empty directory
<code>rm -r folder/</code>	Remove folder and contents
<code>cp file1 file2</code>	Copy file
<code>mv old new</code>	Move/rename file or folder

Tips

- Use `ls -la` to see all files + metadata.
- Use `tab` for autocomplete.

Practice Questions

1. Create a directory `projects`, go inside it.
 2. Inside it, create a file `notes.txt`, then rename it to `summary.txt`.
 3. Delete the file and directory in one line.
-

👉 Section 2: File Viewing & Editing

😬 View Commands

Command	Purpose
<code>cat file.txt</code>	Show contents
<code>less file.txt</code>	Paginated view
<code>head file.txt</code>	First 10 lines
<code>tail file.txt</code>	Last 10 lines
<code>tail -f log.txt</code>	Live updates

😬 Editing Commands

Command	Purpose
<code>nano file.txt</code>	Simple terminal editor
<code>vim file.txt</code>	Advanced editor
<code>code .</code>	Open VS Code (if installed)

🐱 Practice Questions

1. View the last 5 lines of a log file.
2. Open a file using nano, write text, save and exit.
3. Use `cat` to combine two files into a third one.

👉 Section 3: Permissions & Ownership

😬 Permission Syntax:

```
-rw-r--r-- 1 user group size date file
```

Type	User	Group	Others
r = read	r	r	r
w = write	w	-	-
x = execute	-	-	-

😬 Change Permissions

```
chmod +x script.sh    # Add execute
chmod 755 script.sh    # rwxr-xr-x
```

😬 Change Ownership

```
sudo chown newuser:newgroup file
```

🐱 Practice Questions

1. Give a file executable permission.
2. Remove write access for group and others.
3. Change file owner to a different user (use sudo).

👉 Section 4: Searching & Filtering

😬 Key Commands

Command	Use
<code>find . -name "*.js"</code>	Find JS files
<code>grep "TODO" file.js</code>	Find pattern in file
<code>grep -r "TODO" .</code>	Recursively grep
<code>find . -size +1M</code>	Find files larger than 1MB
<code>find dir -name "*.js" -exec grep "TODO" {} \;</code>	Combined find + grep

🐱 Practice Questions

1. Find all `.py` files in current dir and subdirs.
2. Search for the word "import" in all `.js` files.
3. List all files larger than 100MB.

👉 Section 5: Input & Output Redirection

😬 Redirection

Symbol	Function
<code>></code>	Write (overwrite)
<code>>></code>	Append
<code><</code>	Read from file
<code> </code>	Pipe to another command
<code>tee</code>	Write to file and screen

😬 Examples

```
echo "hello" > hello.txt  
cat file.txt | grep "error" | tee errors.txt  
wc -l < file.txt
```

🐱 Practice Questions

1. Append "hello world" to a file.
 2. Count lines in a file using redirection.
 3. Pipe `ps aux` to grep for "chrome".
-

👉 Section 6: Process Management

😬 Job Control

Command	Description
<code>ps aux</code>	Show all processes
<code>kill PID</code>	Kill a process
<code>kill -9 PID</code>	Force kill
<code>top</code>	Real-time process viewer
<code>fg</code>	Bring job to foreground
<code>bg</code>	Resume job in background
<code>jobs</code>	List background jobs
<code>&</code>	Run in background
<code>Ctrl+Z</code>	Pause a process
<code>Ctrl+C</code>	Kill foreground job

🐱 Practice Questions

1. Start a long-running job in background.
 2. Pause a foreground process and bring it back.
 3. Kill a process by name using `pkill`.
-

👉 Section 7: Combining Commands

😬 Operators

Symbol	Description
<code>;</code>	Run commands sequentially regardless of result
<code>&&</code>	Run second command only if the first succeeds
<code>,</code>	<code>,</code> Run second command only if the first fails

😬 Examples

```
echo "Hello"; echo "World"
mkdir newdir && cd newdir
false || echo "This runs only if false fails"
```

🐱 Practice Questions

1. Chain 3 commands using `;`.
2. Only run `echo Done` if a directory creation succeeds.
3. Try a failing command and handle it using `||`.

👉 Section 8: Networking Commands

😬 Commands

Command	Purpose
<code>ping google.com</code>	Check host reachability
<code>curl -I example.com</code>	Fetch headers
<code>wget url</code>	Download files
<code>nslookup domain.com</code>	Get DNS info
<code>tracert google.com</code>	Show packet path
<code>netstat -tuln</code> / <code>ss -tuln</code>	Show open ports
<code>ip a</code>	Show IP addresses

🐱 Practice Questions

1. Use `ping` to check a website's status.
2. Fetch HTTP headers from `example.com`.

3. Find your own IP address with `ip a`.

👉 Section 9: Bash Scripting Basics

😬 Structure

```
#!/bin/bash
echo "Hello $USER"
```

😬 Features

- Variables: `name="Partha"`
- Input: `read var`
- Conditions: `if [$x -gt 10]; then ... fi`
- Loops: `for`, `while`
- Args: `$0`, `$1`, ...

🐱 Practice Questions

1. Create a script to greet the user by name.
 2. Write a loop that counts 1 to 5.
 3. Write a script that takes 2 numbers and adds them.
-

👉 Section 10: Package Management

😬 Commands

Command	Purpose
<code>sudo apt update</code>	Refresh package lists
<code>sudo apt upgrade</code>	Upgrade installed packages
<code>sudo apt install</code>	Install a package
<code>sudo apt remove</code>	Remove a package
<code>dpkg -i file.deb</code>	Install local package
<code>apt search name</code>	Search packages
<code>apt show name</code>	Show package details

🐱 Practice Questions

1. Install a package (`curl`) and remove it.
 2. Search for packages with "python".
 3. Install a `.deb` package and fix broken dependencies.
-

👉 Section 11: Environment Variables

😬 Common Variables


Variable	Description
<code>\$PATH</code>	Directories where shell looks for executables
<code>\$HOME</code>	User home dir
<code>\$PWD</code>	Current directory
<code>\$USER</code>	Username

😬 Commands

```
export VAR="value"    # Set temporary var
unset VAR             # Remove var
source ~/.bashrc      # Reload shell
```

🐱 Practice Questions

1. Set and echo a custom variable.
2. Add a directory to `$PATH`.
3. Make a variable permanent by editing `.bashrc`.

 **Congratulations!** You've completed 11 core terminal sections. This guide is now your command-line Bible.

Next steps: Advanced Bash scripting, automation, server management, or Git internals! 