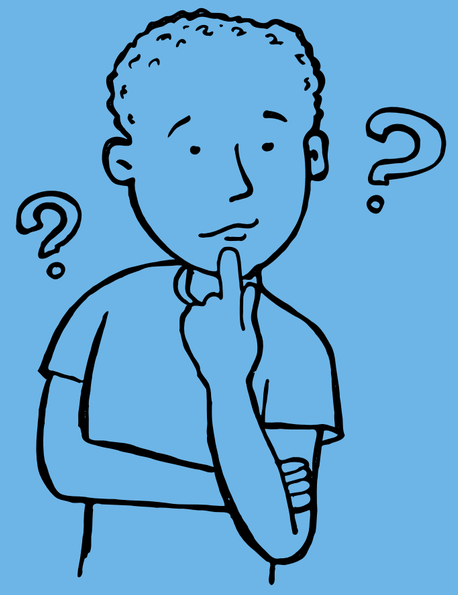
CSTP:1206

Introduction to Internet Programming & Web Applications

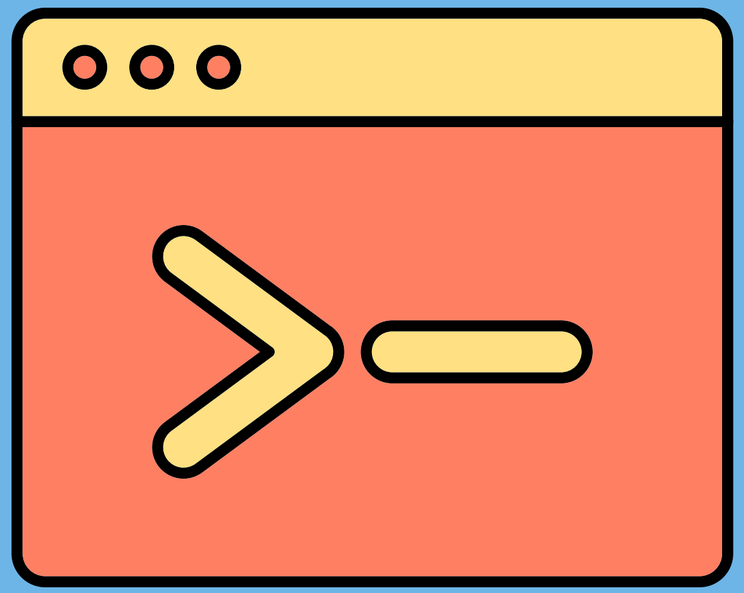
Command Line Basics

Prabhjyot Gambhir
(PRABH)

Why Command Line ?



Command Line Interface (CLI)



The majority of the elements required for user interaction are provided by the graphical user interface, or GUI, which is created for a general user. Due to its consumer-oriented design, your smartphone lacks a user-accessible command-line interface, or CLI.

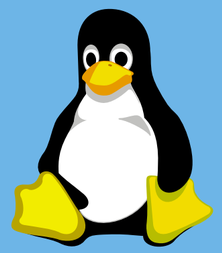
You, however, are not a typical consumer. You work in programming. Everything you can do with the GUI and more can be done with the CLI.

One of the best aspects of the CLI is that you can easily automate routine operations by converting those simple commands into scripts that you can run. Because of this, several well-known programming frameworks and languages, including Node.js, Ember, React, Elixir, Ruby, Python, and many others, largely rely on the CLI.

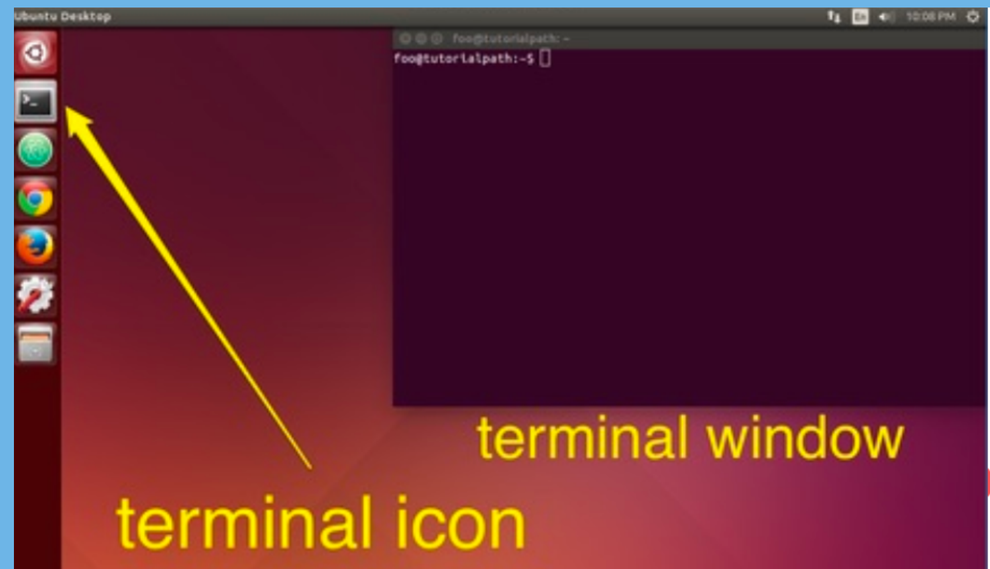
Running a Terminal

To run a command-line command, we first need to start a terminal, which is the program that gives us a command line. The exact details depend on the particular operating system you're using.

On macOS, you can open a terminal window using the macOS application Spotlight, which you can launch either by typing ⌘␣ (Command-space) or by clicking on the magnifying glass in the upper right part of your screen. Once you've launched Spotlight, you can start a terminal program by typing "terminal" in the Spotlight Search bar.



On Linux, you can click the terminal icon as shown on the right.



Running a Terminal

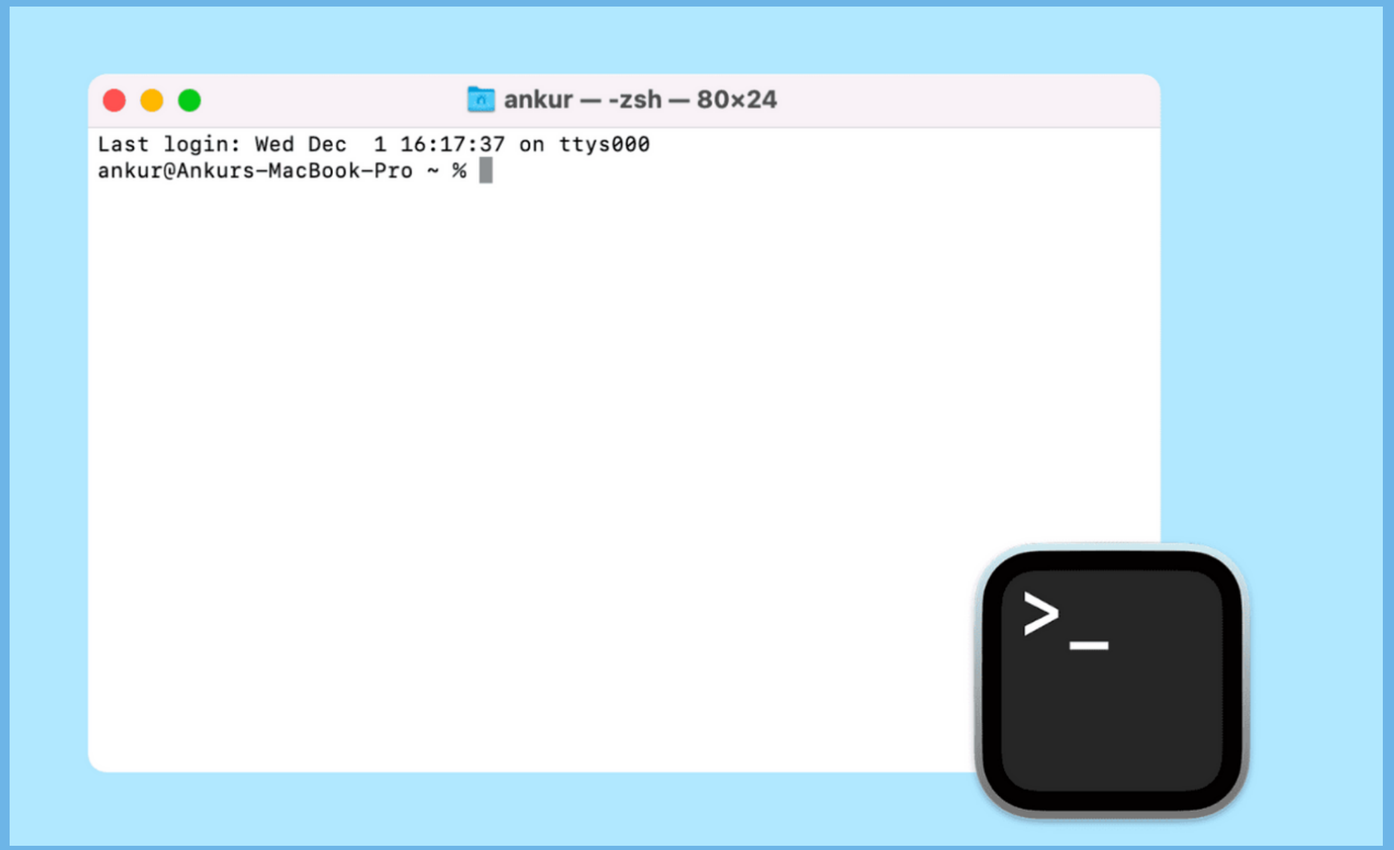
To run a command-line command, we first need to start a terminal, which is the program that gives us a command line. The exact details depend on the particular operating system you're using.

On Windows, the recommended option is to install Linux.
Windows now ships with a working Linux kernel, and you can install any of a number of Linux distributions by following Microsoft's own instructions.

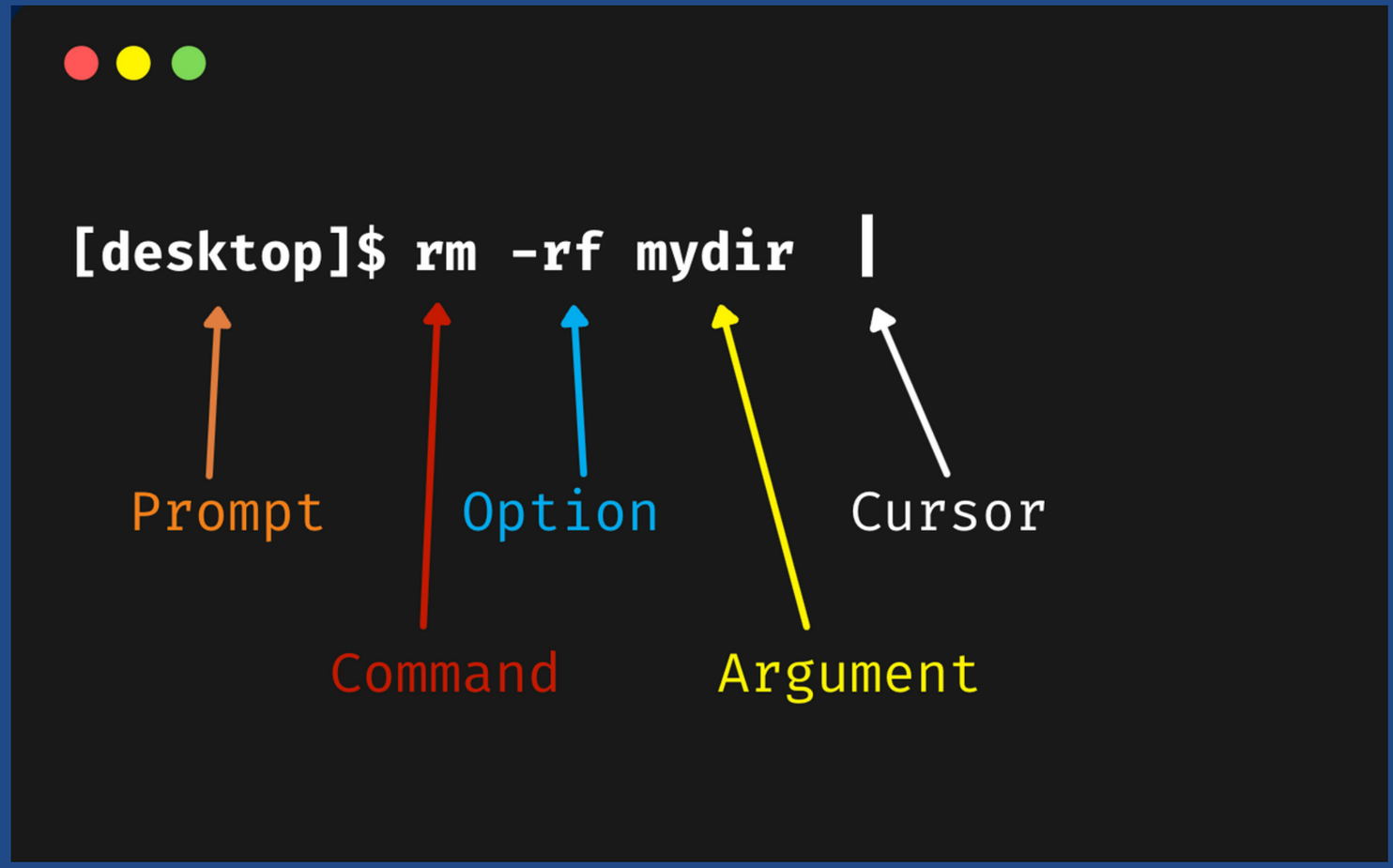
Alternatives:

- **Hyper**
- **Git-bash**

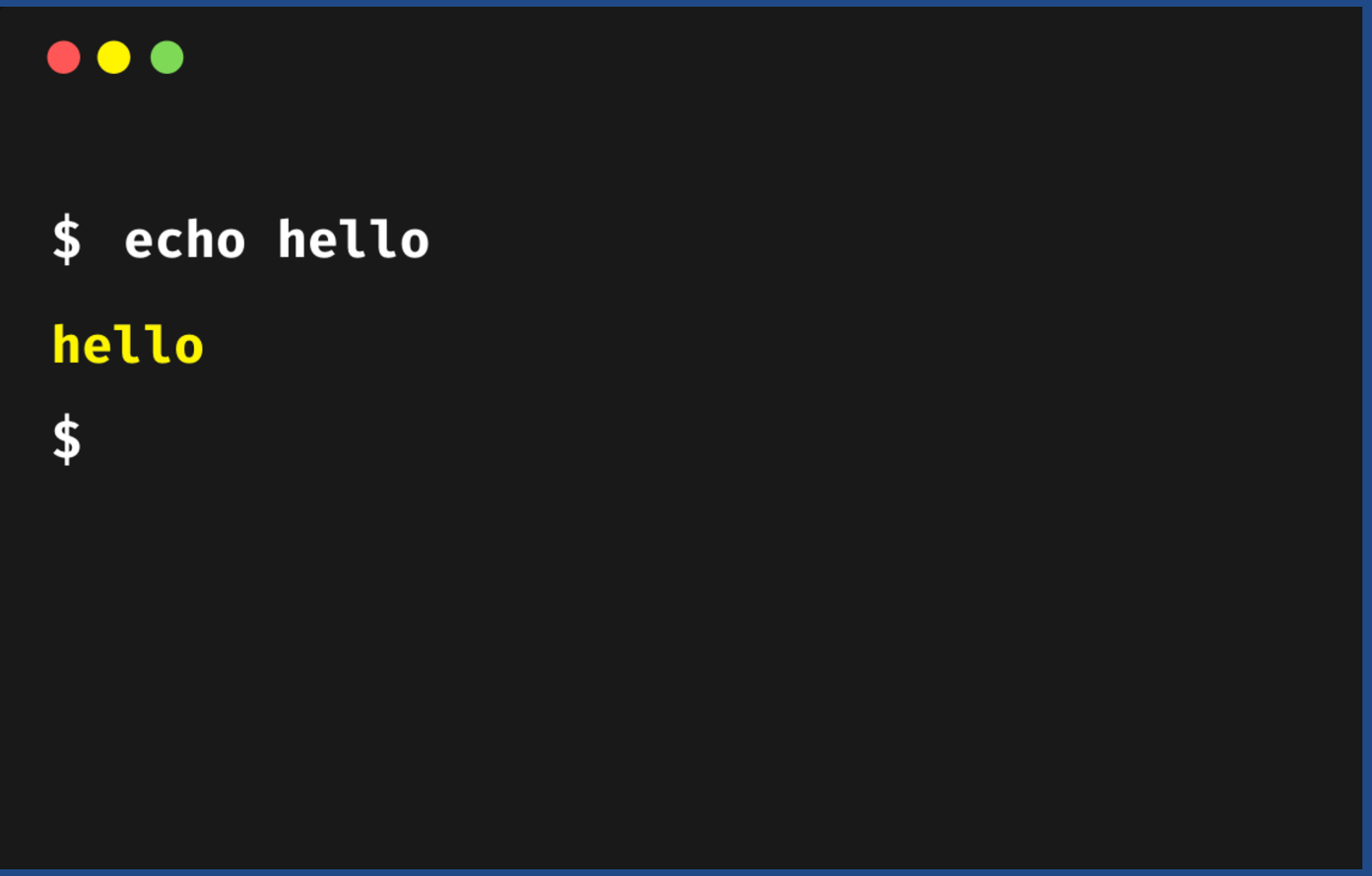
No matter what operating system you are using, your terminal window should resemble the one below, but specifics may vary.



The first symbol on every command line is intended to "prompt" you into action. The prompt is typically preceded by information that relies on the specifics of your system and concludes with a dollar symbol ($) or a percent sign%.



Our First Command



The command we want to use is echo, and the parameter is the string of characters that we want to print. When prompted, write "echo hello" and press the Return key (also known as Enter) to execute the echo command:

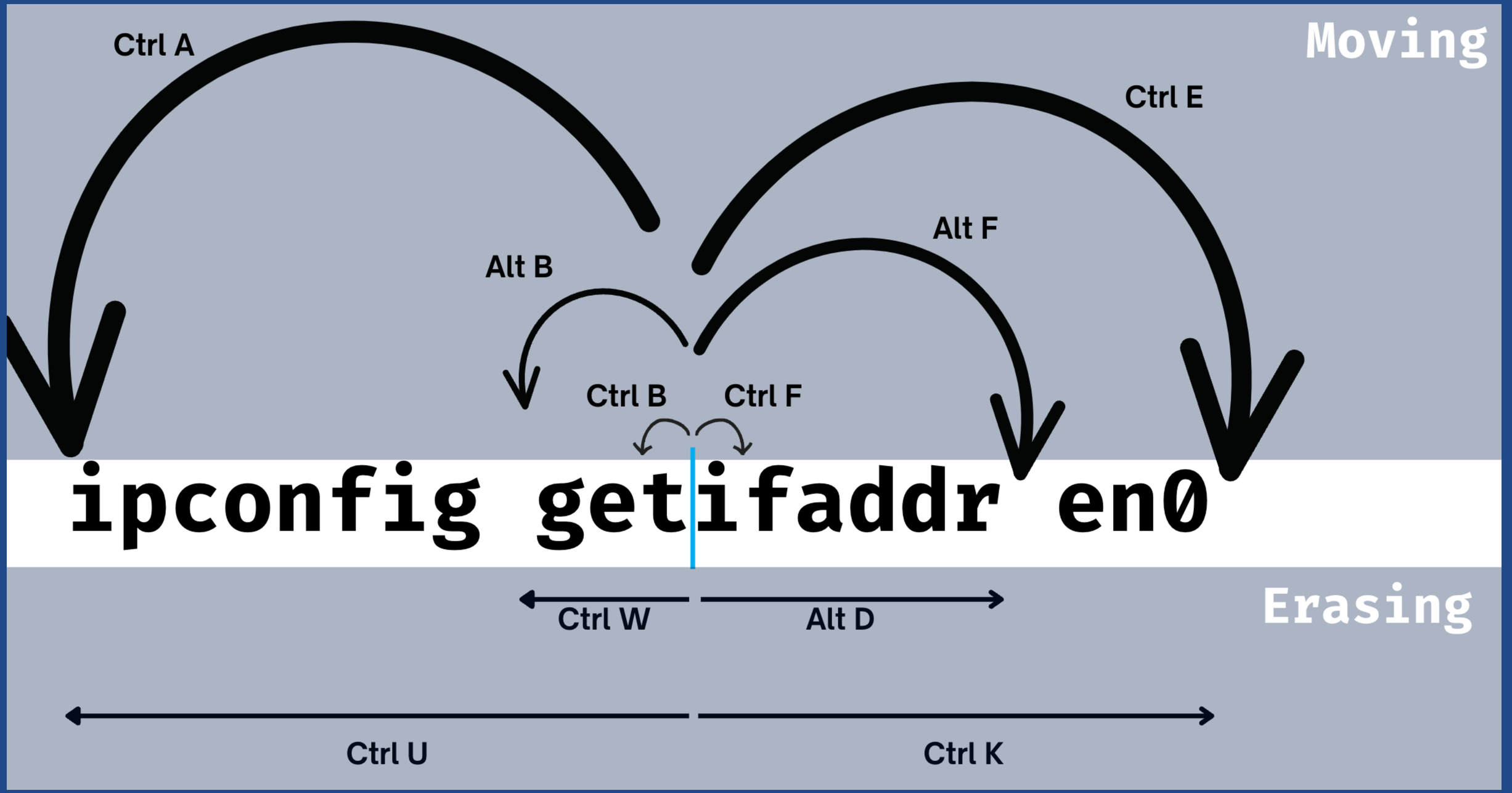
Our Second Command

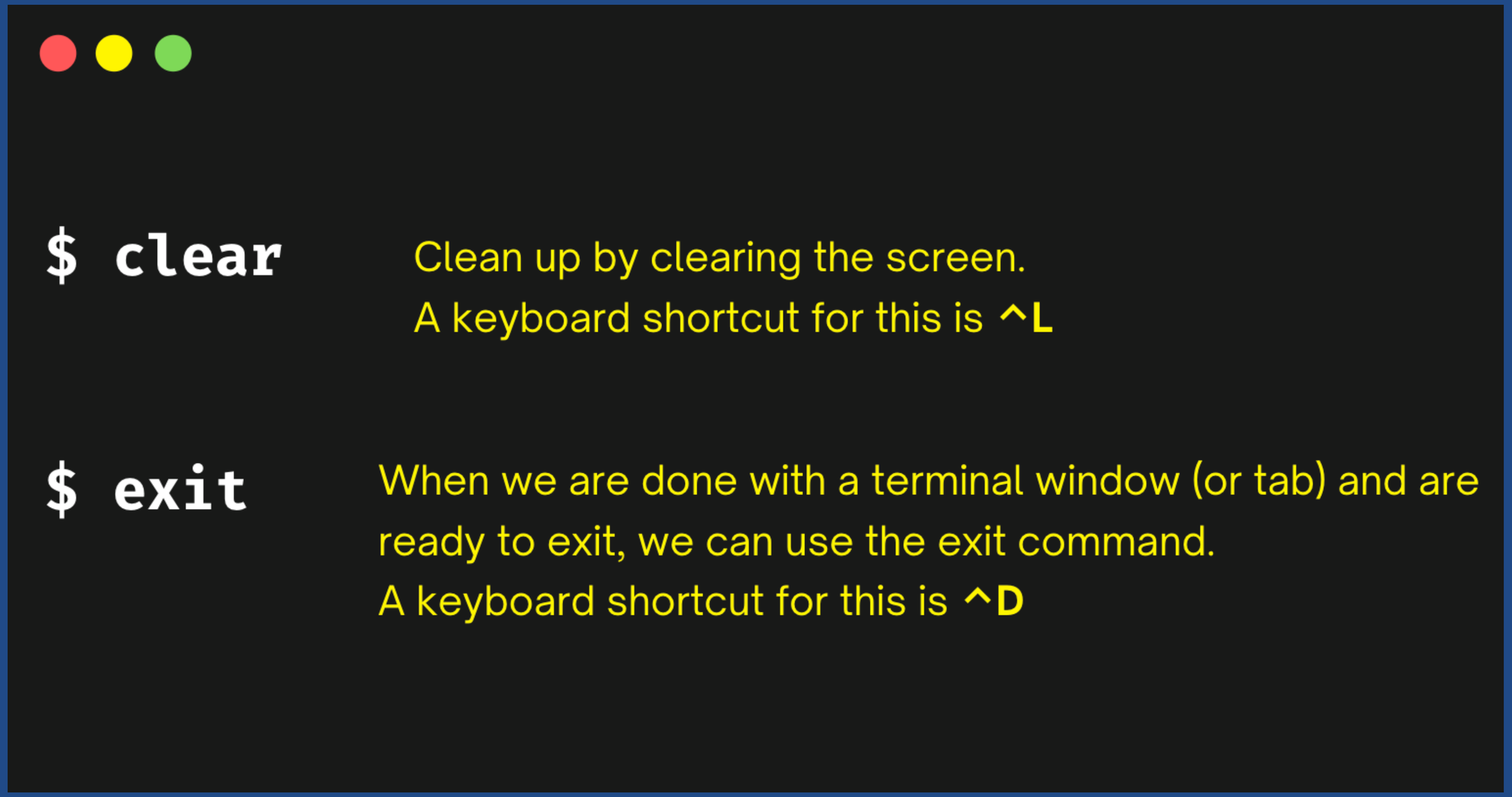
```
ECHO(1)                    General Commands Manual                   ECHO(1)

NAME
     echo – write arguments to the standard output

SYNOPSIS
     echo [-n] [string ...]

DESCRIPTION
     The echo utility writes any specified operands, separated by single blank
     (' ') characters and followed by a newline ('\n') character, to the
     standard output.

     The following option is available:

     -n    Do not print the trailing newline character.  This may also be
           achieved by appending '\c' to the end of the string, as is done by
           iBCS2 compatible systems.  Note that this option as well as the
           effect of '\c' are implementation-defined in IEEE Std 1003.1-2001
           ("POSIX.1") as amended by Cor. 1-2002.  Applications aiming for
           maximum portability are strongly encouraged to use printf(1) to
           suppress the newline character.

:
```

To learn more about available commands. We use a command-line command called man (short for "manual"), and we use it by typing man and then the name of the command we want to learn more about:

SOME SHORTCUT KEYS

Key	Symbol
Command	⌘
Control	^ / CTRL
Shift	⇧
Option	⌥ / ALT
Up, down, left, right	↑ ↓ ← →
Enter/Return	↵
Tab	⇥
Delete	⌫


Moving
Ctrl A
Ctrl E
Alt F
Alt B
Ctrl B
Ctrl F
ipconfig getifaddr en0
Erasing
Ctrl W
Alt D
Ctrl U
Ctrl K



You can also use Cmd + K to Clear the screen on macbook

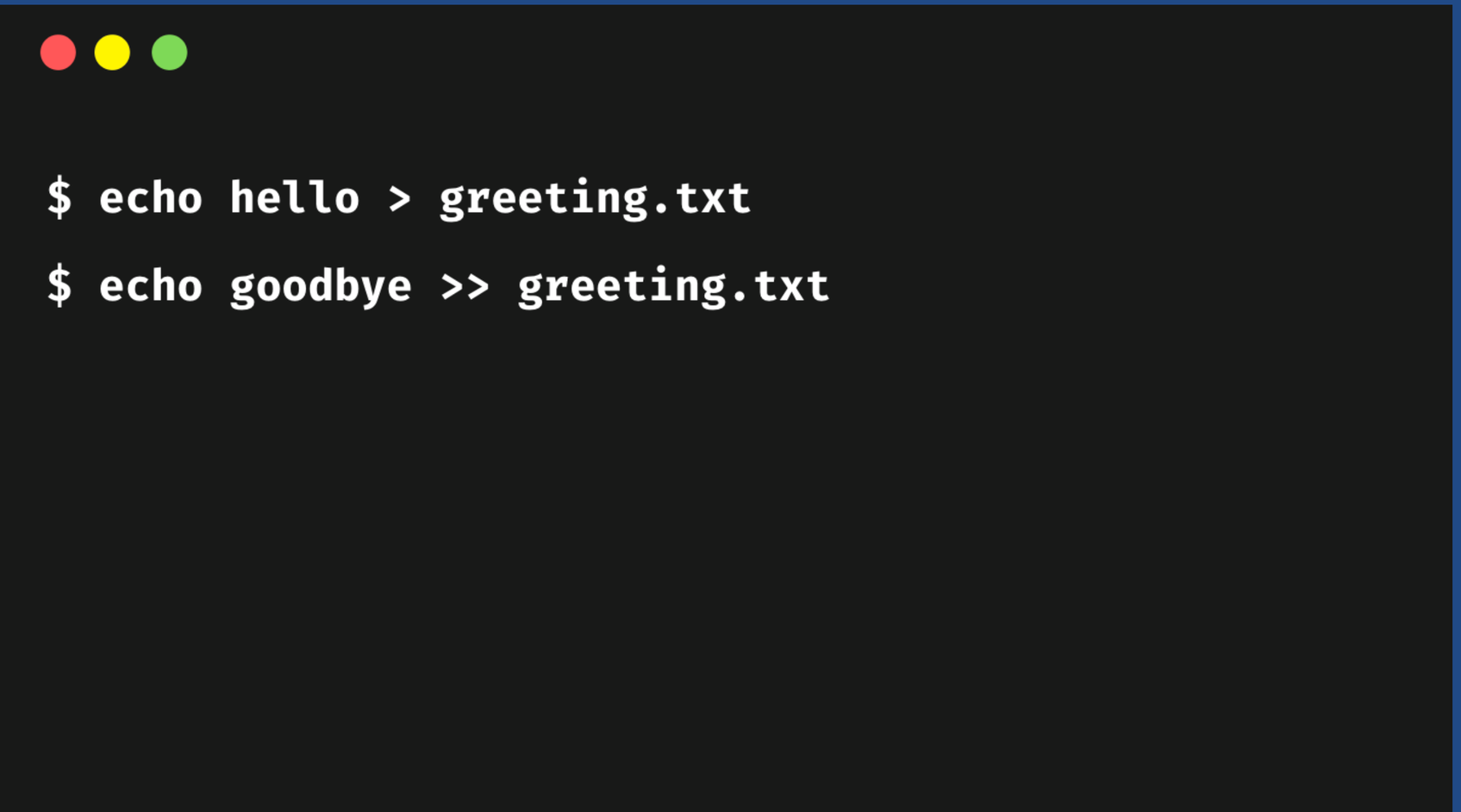
Command Summary

Command	Description	Example
echo <string>	Print string to screen	$ echo hello
man <command>	Display manual page for command	$ man echo
^C / Ctrl + C	Get out of trouble	$ tail ^C
^A / Ctrl + A	Move to beginning of line	
^E / Ctrl + E	Move to end of line	
^U / Ctrl + U	Delete to beginning of line	
Option-click	Move cursor to location clicked	
Up & down arrow	Scroll through previous commands	
clear or ^L / Ctrl + L	Clear screen	$ clear
exit or ^D / Ctrl + D	Exit terminal	$ exit

Adding and Appending Content

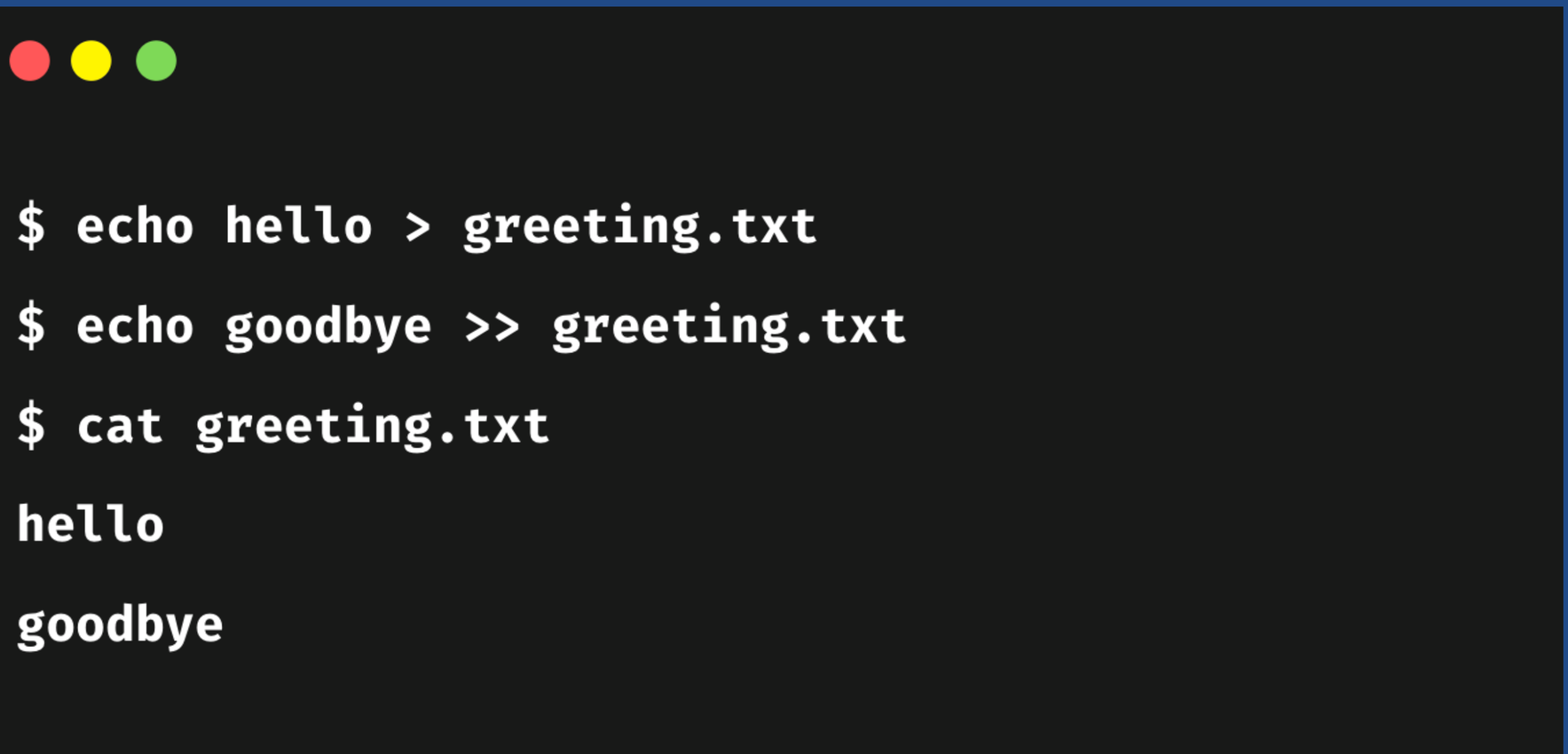
> takes the string output from echo and redirects its contents to a file.

>> appended the string from echo to the file



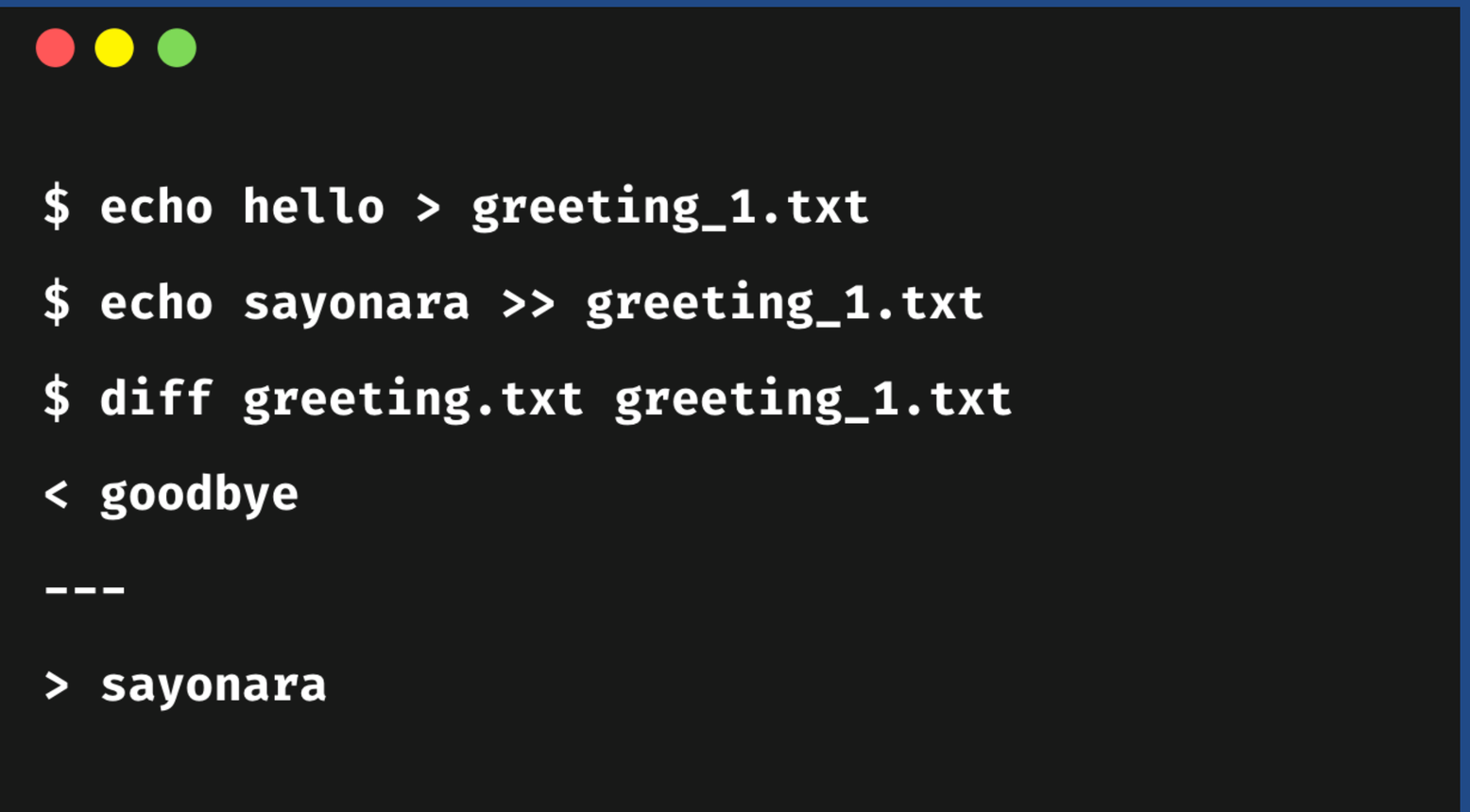
Cat Command

The word "concatenate," which is shortened to "cat," is used to display the contents of a single file.



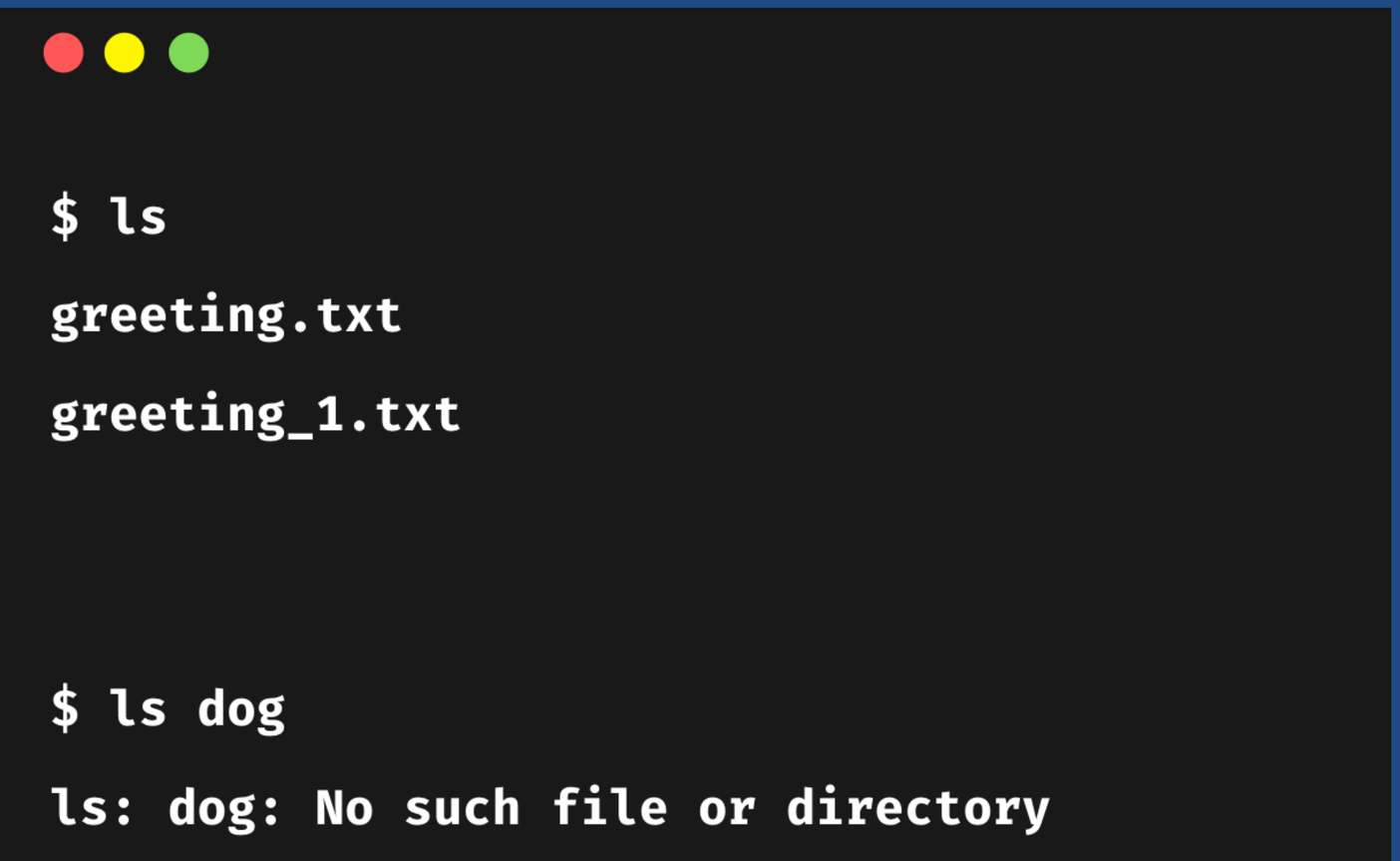
Diff Command

Unix systems include the helpful diff function to make it easier to compare files that are similar but not identical.



Listing

The ls command simply lists all the files and directories in the current directory (except for those that are hidden)



The ls command can be used to determine whether a file (or directory) is present because attempting to ls a nonexistent file results in an error message reading "No such file or directory."

Hidden Files

Hidden files (and directories) are a feature of Unix that aren't always visible when listing files. Starting with a dot identifies hidden files and folders. and are frequently employed to store user preferences, among other things.

```
$ echo "text in hidden file" > .gitignore
$ cat .gitignore
text in hidden file
$ ls
greeting.txt
greeting_1.txt
$ ls -a
.gitignore
greeting.txt
greeting_1.txt
```

To get ls to display hidden files and directories, we need to pass it the -a option (for “all”)

RENAMING A FILE

We can rename a file using mv command

```
$ echo "test text" > first_.txt
$ mv first_.txt first_test.txt
$ ls
first_test.txt
```

Copying A FILE

The way to copy a file is with cp, short for “copy”

```
$ cp first_test.txt second_test.txt
$ ls
second_test.txt
test_file.txt
```

Deleting A FILE

The way to delete a file is with rm, short for “remove”

```
$ rm second_test.txt
$ ls second_test.txt
ls: second_test.txt: No such file or directory
```

Command Summary

Command	Description	Example
>	Redirect output to filename	$ echo foo > foo.txt
>>	Append output to filename	$ echo bar >> foo.txt
cat <file>	Print contents of file to screen	$ cat hello.txt
diff <f1> <f2>	Diff files 1 & 2	$ diff foo.txt bar.txt
ls	List directory or file	$ ls hello.txt
ls -l	List long form	$ ls -l hello.txt
ls -rtl	Long by reverse modification time	$ ls -rtl
ls -a	List all (including hidden)	$ ls -a
touch <file>	Create an empty file	$ touch foo
mv <old> <new>	Rename (move) from old to new	$ mv foo bar
cp <old> <new>	Copy old to new	$ cp foo bar
rm <file>	Remove (delete) file	$ rm foo
rm -f <file>	Force-remove file	$ rm -f bar

Do Give it a try!

https://cmdchallenge.com/