Orientation — Shell & Getting Help (Core)

Linux Commands Course · Section 0

IDSchool

Terminal vs Shell

A **terminal** is the window where you type. A **shell** is the program that reads what you type and runs it (e.g., bash, zsh, fish).

You talk to the OS through the shell. This course uses a Bourne-style shell (bash/zsh).

Which shell am I using?

Print the value of the SHELL environment variable:

```
echo $SHELL
```

Typical outputs: `/bin/bash`, `/bin/zsh`.

---

bash and zsh — at a glance

- bash: ubiquitous default on many distros; great for scripts.
- zsh: interactive niceties (completion, prompts) while staying Bourne-compatible for most everyday commands.

You can learn one and be productive in both.

Prompt anatomy

A common prompt looks like this:

```
user@host:~$
```

- user – your account name
- host – machine name
- ~ – your home directory
- $ – normal user (# means root)

Command anatomy

Pattern you’ll see everywhere:

```
command [options] [arguments]
```

Example:

```
echo Hello
```

`echo` is the command; `Hello` is an argument printed to the screen.

---

echo — printing text

Print simple text:

```
echo Hello
```

Preserve spaces by quoting:

```
echo "Multiple words stay together"
```

Show special characters literally by single-quoting:

```
echo 'Use $ and * literally'
```

---

type vs which — what will run?

Discover how the shell resolves a name.

type (shell builtin) tells if something is a builtin, alias, function, or an external program:

```
type echo
```

which searches your PATH and shows the path to an external program:

```
which echo
```

If type says “builtin”, which may print nothing for that name.

Getting help — quick options

Many programs support a short help message:

```
echo --help
```

Bash builtins have builtin help:

```
help echo
```

Use these when you just need a brief synopsis and flags.

Manual pages (man)

Read full documentation for a command:

```
man echo
```

Navigation keys inside man:

- Space / Page Down – next page
- b / Page Up – previous page
- /pattern – search forward
- n / N – next / previous match
- q – quit

---

man sections (concept)

Manuals are grouped into sections (1: user cmds, 5: file formats, 8: admin, etc.).

Open a specific section if names clash:

```
man 1 printf
man 3 printf
```

(Only use if you encounter multiple entries.)

---

whatis and apropos

Show a one-line description for a command name:

```
whatis echo
```

Search across man page descriptions by keyword:

```
apropos print
```

Use apropos when you know the task but not the command name.

---

info pages

Some tools use the GNU Info system for their primary docs:

```
info coreutils
```

Navigation: Space → next, Backspace → previous, q → quit.

Session hygiene — history

List your recent commands with line numbers:

```
history
```

Press ↑ or ↓ to scroll through previous commands at the prompt. You can re-edit and re-run them quickly.

Session hygiene — clear & reset

Clear the visible screen contents:

```
clear
```

If your terminal display gets garbled (binary noise, weird characters), re-initialize it:

```
reset
```

`reset` is safe; it just redraws and resets modes.

Exit the shell

End the current shell session:

```
exit
```

Keyboard shortcut: **Ctrl+D** (sends End-Of-File to the shell).

Keyboard shortcuts (must-know)

Shortcut	What it does
Ctrl+C	Stop current running command
Ctrl+D	Exit shell or end input line
Ctrl+L	Clear screen (like `clear`)
↑ / ↓	Browse command history
Tab	Auto-complete names

Summary

- Shell: the interpreter you talk to (`bash`, `zsh`)
- Identify commands with `type` / `which`
- Learn quickly via `--help`, `help`, `man`, `whatis`, `apropos`, `info`
- Keep sessions tidy with `history`, `clear`, `reset`
- Exit cleanly with `exit` or **Ctrl+D**