COMMAND LINE CHEATSHEET

$ _

Afterhours Spring '26 • Week 1 • Sarah Grant

⌂ NAVIGATION

pwd
Print working directory — where am I?

ls
List files and folders in current directory

ls -la
List ALL files (including hidden) with details

cd <folder>
Change directory — go into a folder

cd ..
Go up one level

cd ~
Go to your home directory

cd /
Go to the root of the file system

TIP: Use Tab to autocomplete file and folder names. Press Tab twice to see all options.

☰ FILE OPERATIONS

touch <file>
Create an empty file

mkdir <folder>
Create a new folder

mkdir -p a/b/c
Create nested folders in one go

cp <src> <dest>
Copy a file

cp -r <src> <dest>
Copy a folder and everything in it

mv <src> <dest>
Move or rename a file/folder

rm <file>
Delete a file (no undo!)

rm -r <folder>
Delete a folder and everything in it

WARNING: rm has no trash can. Deleted files are gone forever. Double check before you press Enter.

▶ READING FILES

cat <file>
Print entire file contents to screen

less <file>
Scroll through a file (q to quit)

head -n 10 <file>
Show first 10 lines

tail -n 10 <file>
Show last 10 lines

wc -l <file>
Count lines in a file

🔒 PERMISSIONS & USERS

```
$ ls -la
-rwxr-xr--  1  sarah  staff  hello.sh
 owner    group    other
r=read  w=write  x=execute
```

chmod +x <file>
Make a file executable (so you can run it)

chmod 755 <file>
Owner: full access, others: read + execute

chown <user> <file>
Change who owns a file

sudo <command>
Run a command as the superuser (root)

whoami
Who am I logged in as?

TIP: sudo means "superuser do". Use it when you need admin access. Be careful with sudo rm!

⚙ ENVIRONMENT

echo $PATH
Show where your system looks for programs

echo $HOME
Show your home directory path

export MY_VAR='hello'
Set an environment variable

env
Show all environment variables

which <program>
Find where a program lives on your system

history
Show your recent commands

TIP: Press the Up arrow key to cycle through previous commands. Ctrl+R lets you search your history.

? GETTING HELP

man <command>
Open the manual page for any command

<command> --help
Quick help / usage info

type <command>
Find out what a command is

⌨ ESSENTIAL SHORTCUTS

Ctrl + C Cancel / stop current command

Ctrl + D Exit / close terminal

Ctrl + L Clear the screen

Ctrl + A Jump to start of line

Ctrl + E Jump to end of line

Tab Autocomplete

↑ / ↓ Previous / next command

COMMAND LINE CHEATSHEET

$ _

Afterhours Spring '26 • Week 1 • Sarah Grant

FINDING THINGS

`find . -name '*.txt'`
Find all .txt files from current directory down

`find /home -type d`
Find only directories

`grep 'word' <file>`
Search for text inside a file

`grep -r 'word' <folder>`
Search for text in all files in a folder

`grep -i 'word' <file>`
Case-insensitive search

PIPES & REDIRECTION

`command1 | command2`
Send output of one command into another

`command > file.txt`
Save output to a file (overwrites!)

`command >> file.txt`
Append output to a file

`command 2> errors.txt`
Save error messages to a file

```
# Example: find all errors in a log file,
# sort them, count unique ones
$ cat server.log | grep 'error' | sort | uniq -c
```

TIP: Pipes are powerful! Think of them as assembly lines — each command does one job and passes the result along.

YOUR FIRST SCRIPT

```
#!/bin/bash

# My first script!
echo "Hello, I am $(whoami)"
echo "Today is $(date)"
echo "I'm in directory $(pwd)"
echo "Files here:"
ls -la
```

1. Save as **`hello.sh`** using a text editor (nano hello.sh)
2. Make executable: **`chmod +x hello.sh`**
3. Run it: **`./hello.sh`**

TIP: The #!/bin/bash line (called "shebang") tells the system to use bash to run the script.

TEXT EDITORS IN TERMINAL

nano — beginner-friendly, shows controls at bottom

`nano <file>` Open file for editing

`Ctrl + O` Save the file

`Ctrl + X` Exit nano

`Ctrl + K` Cut a line

`Ctrl + U` Paste a line

vim — powerful but has a learning curve

`i` Enter insert (typing) mode

`Esc` Return to command mode

`:w` Save

`:q` Quit (:q! to force quit)

`:wq` Save and quit

TIP: Start with nano. It shows you the keyboard shortcuts right on screen. Move to vim later if you want more power.

SYSTEM INFO

`df -h`
Show disk space usage (human-readable)

`du -sh <folder>`
Size of a folder

`top`
Live view of running processes (q to quit)

`ps aux`
List all running processes

`uname -a`
Show system information

`date`
Current date and time

`uptime`
How long the system has been running

NETWORKING BASICS (PREVIEW)

`ping <host>`
Test if you can reach a server

`curl <url>`
Fetch a web page from the command line

`wget <url>`
Download a file from the web

`ifconfig / ip addr`
Show your network interfaces

`ssh user@host`
Connect to a remote machine

We'll go deeper into networking when we set up our own servers in Weeks 3-4!

ANATOMY OF A COMMAND