

CORE LINUX/BASH COMMANDS			
COMMAND	PURPOSE	EXAMPLE	NOTES
pwd	Show where you currently are (print working directory)	pwd	Bash stands for Bourne Again Shell. It's a command-line interface (CLI) — you type instructions instead of clicking buttons. It's the default shell on most Linux systems (and macOS), and it's available on Windows through WSL or Git Bash. Essentially, Bash is the program that reads your commands and tells the computer what to do.
ls	List files and folders in the current location	ls	
ls -l	List with details (permissions, size, dates)	ls -l	
cd	Change directory	cd /home/user	
cd ..	Go up one folder	cd ..	Helpful Notes Commands are short because they were invented when people typed on slow teletype machines. Most commands follow the pattern: command [options] target EXAMPLE: ls -l /home/nathaniel You can press TAB to auto-complete filenames. You can press ↑ and ↓ arrow keys to scroll through previous commands.
mkdir	Make a new directory (folder)	mkdir project	
touch	Create a new empty file	touch notes.txt	
rm	Remove a file (delete)	rm old.txt	
rm -r	Remove a folder and everything inside	rm -r myfolder	
cp	Copy a file	cp file.txt backup.txt	
mv	Move or rename a file/folder	mv file.txt archive/	
cat	Show the contents of a file	cat log.txt	
less	View a large file one page at a time	less bigfile.log	
echo	Print text or send text into a file	echo "hello" > file.txt	
grep	Search for text inside files	grep "error" log.txt	
chmod	Change file permissions	chmod +x script.sh	
chown	Change file owner	sudo chown user:user file.txt	

sudo	Run as administrator/root	sudo apt update
history	List previous commands	history
clear	Clear the terminal screen	clear
exit	Leave the terminal or close a session	exit

BASH FILE TEST FLAGS

FLAG	MEANING	WHEN IT'S TRUE	
-d file	Directory	File is a directory	
-e file	Exists	File exists (not always portable)	
-f file	Regular file	File is a normal file (not directory, not device)	
-g file	Group ID set	File has set-group-ID bit	
-r file	Readable	File is readable by current user	
-s file	Non-zero size	File exists and size > 0	
-u file	User ID set	File has set-user-ID bit	
-w file	Writable	File is writable by current user	
-x file	Executable	File is executable	

COMPARING BASH TO PYTHON

CONCEPT	BASH	PYTHON	NOTES
OPERATORS			
Greater than	-gt	>	Bash can also use the simple symbols like python
Less than	-lt	<	
Equal too	-eq	==	
Not Equal	-ne	!=	
Greater than or equal to	-ge	>=	
Less than or equal to	-le	<=	
IF - ELSE BLOCKS			
Print output	echo	print()	
If block starts	if ...; then	if ...:	
Else-if	elif ...; then	elif ...:	
End if-else block	fi	indentation ends block	
Comparisons	-gt -lt -eq	> < ==	
Curly braces	none	none	
Indentation	optional	required	

NAMING CONVENTIONS

RULE	BASH	PYTHON	
lowercase with underscores			
no spaces around =			
\$var to read			
can start with letter or _			
cannot start with number			
underscores allowed			
hyphens allowed			
uppercase for constants	common	common	

CASE BLOCK == INPUT BLOCK

RULE	BASH	PYTHON	
Prompt for input	read -p "?" ANSWER	input("?")	
Start branch block	case "VAR" in	if, elif, match	
Single-choice matching	[yY])	if var in ('y', 'Y'):	
Multiple pattern matching	[yY][eE][sS])	if var.lower() == "yes":	
Default / fallback	*)	else:	
Close block	esac	indentation ends block	

BASH SYNTAX ELEMENT	MEANING	EXAMPLE	
read -p "Prompt" VAR	Shows a prompt and stores user input in VAR	read -p "Name: " NAME	
case "\$VAR" in	Begin case block using value of VAR	case "\$ANSWER" in	
[yY])	Match single character Y or y	Accepts: Y, y	
[yY][eE][sS])	Match full word case-insensitive	Accepts: yes, YES, YeS, etc.	
command	Action to perform when match occurs	echo "You can have a beer :)"	
::	End of this case option	Ends the match section	
*)	Default / catch-all	If nothing else matched	
esac	Ends the entire case block	Reverse of case	

	NOTES		
	NOTES		

[illegible]