#_ Essential SHELL SCRIPTING Operations [+100]

Basic Shell Scripting Commands:

- echo: Display text or variables.
- cd: Change the current directory.
- pwd: Print the current working directory.
- ls: List files and directories.
- touch: Create an empty file.
- mkdir: Create a new directory.
- rmdir: Remove an empty directory.
- rm: Remove files or directories.
- cp: Copy files or directories.
- mv: Move or rename files/directories.
- cat: Concatenate and display file contents.
- more: Display file contents page by page.
- less: Display file contents interactively.
- head: Display the beginning of a file.
- tail: Display the end of a file.
- grep: Search text using patterns.
- find: Search for files and directories.
- wc: Count lines, words, and characters.
- sort: Sort lines in a file.
- cut: Remove sections from lines of files.
- sed: Stream editor for text manipulation.
- awk: Text processing tool.
- tee: Redirect output to multiple files.
- chmod: Change file permissions.
- chown: Change file ownership.
- ps: List running processes.
- top: Display dynamic process information.
- kill: Terminate processes.
- df: Show disk space usage.
- du: Estimate file and directory space usage.
- date: Display or set the date and time.

By: Waleed Mousa

- cal: Display a calendar.
- tar: Archive files.
- zip: Compress files.
- unzip: Extract files from a ZIP archive.
- curl: Transfer data with URLs.
- wget: Download files from the internet.
- ping: Test network connectivity.
- ifconfig: Display or configure network interfaces.
- netstat: Network statistics.
- hostname: Show or set system hostname.
- who: Display who is logged in.
- users: List users currently logged in.
- groups: List user groups.
- passwd: Change user password.
- su: Switch user.
- sudo: Execute commands with superuser privileges.
- exit: Exit the shell or log out.
- alias: Create command shortcuts.
- history: Display command history.
- !: Repeat previous command.
- source: Execute a script in the current shell.
- ./script.sh: Execute a shell script.
- #: Add comments in scripts.
- $: Access and manipulate variables.
- export: Make variables available to subshells.
- read: Read input from the user.
- for loop: Iterate over a list.
- while loop: Execute commands while a condition is true.
- case: Execute commands based on pattern matching.
- test: Evaluate conditions.
- [ ]: Another form of the test command.
- (( )): Arithmetic evaluation.
- $?: Get the exit status of the last command.
- $0, $1, $2, ...: Access script arguments.
- $$: Get the PID of the current shell.

- $!: Get the PID of the last background command.
- $@: All arguments as separate words.
- $#: Number of arguments.
- $*: All arguments as a single word.

Intermediate Shell Scripting Commands:

- grep -r: Recursively search for text in files.
- find -exec: Execute commands on found files.
- awk scripting: Write more advanced text processing scripts.
- sed scripting: Create complex text transformations.
- cut -f: Specify field delimiters.
- sort -k: Sort using a specific field.
- chmod octal: Set permissions using octal notation.
- chown -R: Recursively change file ownership.
- ps aux: List detailed process information.
- kill -9: Forcefully terminate processes.
- df -h: Display disk space usage in human-readable format.
- du -h: Display disk usage in a more readable format.
- tar -xzvf: Extract compressed tar archives.
- ifconfig eth0 up/down: Enable or disable network interfaces.
- netstat -tuln: List listening ports.
- hostnamectl: View and set system hostname (modern systems).
- whoami: Display the current username.
- id: Show user and group information.
- groups USERNAME: List groups for a user.
- passwd USERNAME: Change another user's password.
- sudo visudo: Edit the sudoers file safely.
- !n: Execute the nth command from history.
- $RANDOM: Generate random numbers in scripts.
- $((expression)): Perform arithmetic operations in scripts.
- if-elif-else: Conditional branching.
- case/esac: Complex case statement.
- $IFS: Internal Field Separator for word splitting.
- $HOME: User's home directory.
- $PATH: System search path for executables.

- $PWD: Current working directory.
- $USER: Current user's name.
- $HOSTNAME: Hostname of the system.
- $SHELL: Current shell.
- $LINENO: Current line number in a script.
- shift: Shift command-line arguments.
- readonly: Make variables read-only.
- trap: Execute commands on signals.
- function: Define reusable functions.
- getopts: Parse command-line options.
- $?: Exit status of the last command.
- $!: Process ID of the last background command.
- $@: All arguments as separate words.
- $#: Number of arguments.
- $*: All arguments as a single word.
- $?: Exit status of the last command.

Advanced Shell Scripting Commands:

- grep -o: Show only matching parts of a line.
- find -type: Search for specific file types.
- awk -F: Specify field separators.
- sed -i: Edit files in-place.
- for loop: Iterate over ranges and patterns.
- while loop: Use conditional loops with complex tests.
- case: Handle complex conditions with case statements.
- eval: Evaluate and execute commands dynamically.
- declare: Create and manipulate variables dynamically.
- select: Create interactive menus.
- set: Set or unset shell options.
- exec: Replace the current shell process.
- read -p: Prompt for input.
- here documents: Pass input to commands.
- process substitution: Use <() and >() to manipulate data.
- I/O redirection: Redirect input and output.
- file descriptors: Work with custom file descriptors.

- pipes: Create pipelines for data processing.
- signals: Handle signals and traps.
- coproc: Run commands in coprocesses.
- arithmetic expansion: Perform complex calculations.
- regex matching: Use regular expressions in scripts.
- string manipulation: Manipulate strings in scripts.
- arrays: Use arrays for data storage.
- associative arrays: Create key-value data structures.
- error handling: Implement error handling in scripts.
- debugging: Debug shell scripts with set -x.
- scripting best practices: Follow scripting best practices.
- shebang: Set the script's interpreter.
- cron: Schedule script execution.
- at: Schedule one-time script execution.
- systemd timers: Create timers for scripts (systemd-based systems).
- journalctl: View system logs (systemd-based systems).
- awk 'BEGIN/END': Execute commands before/after processing.
- awk 'NR': Use the record number in AWK.
- basename: Extract the filename from a path.
- dirname: Extract the directory from a path.
- $FUNCNAME: Get the name of the current function.
- $BASH_SOURCE: Get the name of the script.
- $BASH_VERSION: Get the Bash version.
- readonly -f: Make functions read-only.
- $LINENO: Get the current line number.
- $COLUMNS: Get the terminal's columns.
- $LINES: Get the terminal's lines.
- $RANDOM: Generate random numbers.
- getopts: Handle advanced command-line options.
- $PPID: Get the parent process ID.
- find -print0: Print null-terminated output for safe parsing.
- rsync: Synchronize files and directories.
- curl/wget: Download files from the internet.
- ssh/scp: Securely connect and transfer files.
- expect: Automate interactive tasks.

By: Waleed Mousa

- xargs: Process input arguments.
- printf: Format and print data.
- timeout: Run a command with a time limit.
- logger: Log messages to system logs.
- mktemp: Create temporary files and directories.
- flock: File locking for synchronization.
- lsof: List open files and processes.
- awk 'END': Execute commands after processing input.
- awk 'NF': Filter non-empty lines.
- sed -e: Execute multiple expressions.
- sed -r: Use extended regular expressions.
- shellcheck: Check shell scripts for errors.
- tput: Control terminal text attributes.
- set -e: Exit script on error.
- set -u: Treat unset variables as errors.
- set -o pipefail: Exit on pipeline failure.
- eval "$VAR": Execute dynamic code.
- declare -A: Declare associative arrays.
- $IFS: Input Field Separator (IFS) manipulation.
- read -a: Read input into an array.
- $BASH_ENV: Specify an environment file.
- scp/rsync with SSH keys: Securely transfer files.
- cron with environment variables: Set environment variables in cron jobs.