🐚 Bash Scripting Cheat Sheet

Complete reference guide for Bash scripting syntax, operators, and commands

Connect

Basic Syntax & Operators

SYNTAX	DESCRIPTION	OPERATORS	DESCRIPTION
`#!/bin/bash`	Path for script interpreter	`&&`	Logical AND operator
`bash file.sh`	Execute script in terminal	`||`	Logical OR operator
`./file.sh`	Execute if script is executable	`$#`	Number of arguments passed
`#`	Make comments in script	`$0`	Name of the shell
`$1, $2`	Input parameters	`exit [0-255]`	Exit script with return code
`$()`	Save output of commands	`()`	Run commands in subshell
`$((...))`	Retrieve arithmetic output	`(())`	Arithmetic operations
`${...}`	String manipulation/variable interpolation	`[]`	Filename expansion/string manipulation
`>`	Send output to file	`|`	Run multiple commands together
`>>`	Append output to file	`;`	Separate multiple commands
`~`	Home directory	`<`	Get input from file

File Test Operators & Comparison Operators

FILE TEST OPERATORS	DESCRIPTION	INTEGER & STRING OPERATORS	DESCRIPTION
`-e`	Test if file exists	INTEGER	
`-f`	Test if regular file	`-eq`	Is equal to
`-d`	Test if directory	`-ne`	Is not equal to
`-b`	Test if block device	`-gt`	Is greater than
`-s`	Test if file not zero size	`-ge`	Is greater than or equal to
`-L`	Test if symbolic link	`-lt`	Is less than
`-S`	Test if socket	`-le`	Is less than or equal to
`-r`	Test read permission	STRING	
`-w`	Test write permission	`=`	Is equal to
`-x`	Test execute permission	`!=`	Is not equal to
`-O`	You are owner	`<`	Less than
`f1 -nt f2`	f1 newer than f2	`>`	Greater than
`f1 -ot f2`	f1 older than f2	`-z`	String is null
		`-n`	String is not null

Regular Expressions & Control Structures

REGEX EXPRESSIONS	EXPLANATION	LOOPS & CONDITIONS	EXPLANATION
`.`	Matches any single character	`if then fi`	Test a condition
`?`	Preceding item optional (0 or 1)	`if then else fi`	Test with fallback
`*`	Preceding item 0 or more times	`if then elif else fi`	Multiple condition tests
`+`	Preceding item 1 or more times	`for do done`	Iterate over values
`{N}`	Preceding item exactly N times	`while do done`	Loop while condition true
`{N,}`	Preceding item N or more times	`until do done`	Loop until condition true
`^`	Beginning of line	`sleep time`	Wait specified time
`$`	End of line	`break`	Exit loop, continue script
`[a-d]`	Any character in range a-d	`continue`	Skip to next iteration

Arrays, Functions & System Configuration

BASH ARRAYS	EXPLANATION	SYSTEM CONFIGURATION	EXPLANATION
`array=("elements")`	Create string array	`/etc/profile`	Executed at login
`${array[0]}`	Get first element	`~/.bashrc`	Read by non-login shells
`${array[*]}`	Get all values	TEXT PROCESSING	
`${array[-1]}`	Get last value	`grep foo file.txt`	Search word in file
`${array[@]}`	Expand all elements	`egrep "(foo|bar)" file.txt`	Find multiple words
`shift`	Move $2 to $1	`sed s/foo/bar/g file.txt`	Find and replace text
FUNCTIONS		`locate file.txt`	Quick file location
`function() { content }`	Define a function	PERMISSIONS	
`alias`	List all aliases	`chown user:group file`	Change file ownership
`alias name='command'`	Define an alias	`chmod ugo file.txt`	Change file permissions

Common Utilities & Network Commands

FILE & SYSTEM COMMANDS	EXPLANATION	NETWORK & REMOTE COMMANDS	EXPLANATION
`ls -l`	List by type and permission	`ping host`	Check host reachability
`ls -a`	List all files (including hidden)	`nslookup`	DNS query
`pwd`	Display current directory	`ssh user@host`	Login to remote system
`whoami`	Current logged user	`netstat -ant`	Check network connections
`last`	Display last user logins	`netstat -ent`	Check established connections
`find /home -name *.txt`	Search text files in /home	`ifconfig`	Check network interfaces
`find . -size 10k -print`	Find files >10k current dir	`scp -r dir user@host:/opt/`	Copy to remote recursively
`ps -ef`	Check running services	`scp -r user@host:/opt/ dir/`	Copy from remote recursively
`df -h`	Show filesystem usage	`rsync -avz local user@host:/backup`	Sync files between systems
`du -sh`	Show directory disk usage		
`free -m`	Show memory usage		
`lsof`	List open files by processes		
`top`	Real-time process monitor		
`kill pid`	Kill running process		
`passwd username`	Set/reset user password		

🚀 Bash Scripting Cheat Sheet

Made by M N IQBAL | Follow for more amazing content! 🔥