



Installing Zsh Commands

| | |
|------------------------------------|------------------------------------------------------------------------------------------|
| <code>sudo dnf install zsh</code> | Installs Zsh with user's package manager on Fedora, CentOS and RHEL |
| <code>sudo apt install zsh</code> | Installs Zsh with user's package manager on Debian and Ubuntu |
| <code>sudo port install zsh</code> | Installs Zsh with user's package manager on MacOS and user can install it using MacPorts |
| <code>brew install zsh</code> | Installs Zsh with user's package manager on MacOS and user can install it using |

Navigation Commands

| | |
|------------------------------------|----------------------------------------------------------------------------------------------------|
| <code>cd</code> | Changes the current directory |
| <code>cd ..</code> | Moves up one level in the directory hierarchy |
| <code>cd /</code> | Changes the current directory to the root directory |
| <code>cd ~</code> | Changes the current directory to your home directory |
| <code>cd -P</code> | Changes the current directory and resolves symbolic links |
| <code>cd -L</code> | Changes the current directory without resolving symbolic links |
| <code>cd /path/to/directory</code> | Changes the current directory to the specified directory path |
| <code>code .</code> | Opens the current directory in Visual Studio Code |
| <code>dirs</code> | Displays or manipulates the directory stack |
| <code>explorer .</code> | Opens the current directory in the default file explorer (on Windows) |
| <code>ls</code> | Lists files and directories in the current directory |
| <code>open .</code> | Opens the current directory in the default file explorer (on macOS) |
| <code>pushd</code> | Changes the current directory and pushes the current directory onto the directory stack |
| <code>popd</code> | Removes entries from the directory stack and changes the current directory to the popped directory |

`pwd` Shows the current working directory

Variable & Environment

| | |
|----------------------|----------------------------------|
| <code>declare</code> | Displays or sets shell variables |
| <code>export</code> | Sets environment variables |

Prepared By: [Lamisa Musharrat](#)

Control Flow and Script Execution

| | |
|-----------------------|------------------------------------------------------------------------|
| <code>break</code> | Exits from a loop or switch statement |
| <code>bye</code> | Exits the shell |
| <code>continue</code> | Skips the remaining iterations of a loop and starts the next iteration |
| <code>return</code> | Exits a function with an optional return value |
| <code>exit</code> | Exits the shell with an optional exit status |
| <code>fc</code> | Manages and edits command history |
| <code>shift</code> | Shifts positional parameters to the left by a specified number |
| <code>test</code> | Evaluates conditional expressions and performs tests |
| <code>trap</code> | Sets or displays actions to be taken when a signal is received |

Process Control Commands

| | |
|-------------------------------------|--------------------------------------------------------------------|
| <code>bg [job ...]</code> | Resumes suspended jobs in the background |
| <code>disown [job ...]</code> | Removes jobs from the job table, allowing them to continue running |
| <code>fg [job ...]</code> | Brings jobs to the foreground |
| <code>jobs</code> | Lists active jobs and their statuses |
| <code>kill [options] job ...</code> | Sends signals to processes or job IDs |
| <code>suspend [-f]</code> | Suspends the current shell session |
| <code>wait [job ...]</code> | Waits for background jobs to complete |

Input/Output Commands

| | |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <code>echo</code> | Displays a line of text or the value of a variable on the terminal |
| <code>print</code> | Displays text or variables on the terminal, and various formatting options and can be used to control the output appearance |



readonly Marks variables as read-only

typeset Declares variables with specific attributes

unset Unsets variables or functions

unsetopt Disables shell options

Command Execution and Environment

noglob Executes a command without performing globbing on arguments

rehash Updates the internal hash table of executable commands

[*source*](#) Executes commands from a file in the current shell session

[*time*](#) Displays the system and user times for the current shell session

Command Information and Location

unhash Removes commands from the internal hash table

where Displays all locations where a command is defined

[*which*](#) Displays the path to the executable file of a command

whence Displays information about a command, including its type and location

Shell Built-In Commands

[*alias*](#) Defines or lists aliases for commands

[*unalias*](#) Removes aliases for commands

builtin Executes a built-in shell command

read

Reads input from the user or from a file and assigns it to variables and allows user to prompt for input, store the entered values, and perform further processing based on the Controls terminal settings and behavior, and -fu options force a flush of the terminal output buffer, ensuring that all pending output is displayed immediately

ttctl -fu

Conditional Execution and

if Executes a block of code based on a condition, If the condition is true, the code within the "if"

then Specifies the start of the code block to be executed if the condition of an "if" statement is true

else Specifies the code block to be executed if the condition of the preceding "if" statement is false

fi Marks the end of an "if" statement

do Specifies the start of the code block to be executed in a "for" or "while" loop

done Marks the end of a "for" or "while" loop

for Executes a block of code repeatedly for a specified number of iterations or for each item in a list

until Executes a block of code repeatedly until a certain condition is true

while Executes a block of code repeatedly as long as a certain condition is true

History and Command Editing

fc Views, edits, re-executes, or changes the order of previously entered commands

getln Displays the command history, showing a list of previously executed commands along with their line numbers

getopts optstring Allows to process command-line arguments and options, and enables to create more interactive and flexible shell scripts



Zsh Cheat Sheet



disable Disables shell built-in commands or functions.

enable Enables shell built-in commands or functions

false Returns a non-zero exit status

functions Lists or defines functions

unfunction Removes defined shell functions

hash Displays or modifies command hash table

set Sets or displays shell options or positional parameters

setopt Sets or displays shell options

true Returns a zero exit status

type Displays information about a command

typeset Declares or displays shell variables with additional attributes

unset Unsets shell variables or functions

unsetopt Unsets shell options

Miscellaneous

zcalc Performs arithmetic calculations directly in the shell

xargs Allows users to apply another command to a list of arguments, similar to xargs but with enhanced features

zprof Provides profiling information for Zsh startup files, allowing user to optimize their loading time

zstyle Customizes the behavior and appearance of the shell by setting various options and styles

zargs Applies another command to a list of arguments

zprof Profiles Zsh startup files, helps to optimize their loading time by identifying potential performance bottlenecks

zstyle Configures and customizes various aspects of Zsh's behavior, including shell options, command completion, and highlighting

history

Displays the command history, showing a list of previously executed commands along with their line numbers

File and Directory Management

cp Creates copies of files and directories

file Displays information about the specified file(s)

mkdir Creates a new directory

mkcd Creates a new directory and changes the current directory to it in a single command

mv Moves or renames files and directories

rm Removes a file

rm -r Removes a directory and its contents recursively

touch Creates an empty file or updates the access and modification timestamps of an existing

zmv Allows users to rename multiple files using complex patterns and expressions.

zmvn Allows users to files rename using natural sorting order, which handles numerical sequences in a more human-friendly way

Loading & Unloading Modules

zmodload -dL Loads one or more shared modules into the Zsh shell

zmodload -e Loads one or more modules and generates error messages if loading fails

zmodload [-a [-bcp [-l]]] [-iL] ... Loads modules, specifying various loading options

zmodload -u [-abcdp [-l]] Unloads modules, specifying various unloading options

zmodload -d [-L] [name] Unloads a specific module or all loaded modules

zmodload -d name Unloads a module and its dependencies

zmodload -ud name [dep ...] Unloads a module and its dependencies and unloads any unused modules

zmodload -ab [-L] Lists all available built-in modules



Zsh Cheat Sheet



| | | | |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|---------------------------------------------------------------------------------------------|
| <i>vared</i> | Provides a convenient way to edit variables in the shell using the editor specified by the EDITOR environment variable | <i>zmodload -ab [-i] name [builtin ...]</i> | Loads one or more built-in modules and generates error messages if loading fails |
| <i>push-line</i> | Pushes the current command line onto the buffer and allows you to continue editing it, useful for complex command constructions | <i>zmodload -ub [-i] builtin ...</i> | Unloads one or more built-in modules |
| <i>autoload</i> | Allows users to lazily load Zsh functions | <i>zmodload -ac [-L]</i> | Lists all available conditional modules |
| <i>clone/detach</i> | Allow you to create new instances of the current shell, either as a forked child process (clone) or as a detached background process (detach) | <i>zmodload -ac [-il] name [cond ...]</i> | Loads one or more conditional modules and generates error messages if loading fails |
| <i>zcompile</i> | Compiles Zsh scripts into a more efficient format, which can improve their execution time | <i>zmodload -uc [-il] cond ...</i> | Unloads one or more conditional modules |
| <i>zpty</i> | Provides a way to create and interact with pseudo-terminals from within Zsh | <i>zmodload -ap [-L]</i> | Lists all available parameter expansion modules |
| <i>ztcp</i> | Allows users to create TCP and UDP connections directly from the shell. It can be used for various network-related tasks | <i>zmodload -ap [-i] name [parameter ...]</i> | Loads one or more parameter expansion modules and generates error messages if loading fails |
| <i>zsocket</i> | Allows users to create and interact with Unix domain sockets from within Zsh. This can be useful for inter-process communication | <i>zmodload -up [-i] parameter ...</i> | Unloads one or more parameter expansion modules |
| <i>zparseopts</i> | Simplifies the handling of command-line options and arguments in Zsh scripts by providing a convenient way to parse and process them | <i>zmodload -a [-L]</i> | Lists all available modules |
| <i>zregexparse</i> | Provides a way to parse text using regular expressions and extract specific elements of interest. It is particularly useful for text | <i>zmodload -a [-i] name [</i> | Loads one or more modules and generates error messages if loading fails |
| <i>zsh-mime-setup</i> | Configures default programs and actions associated with different MIME types in the Zsh environment | <i>zmodload -ua [-i] builtin ...</i> | Unloads one or more modules |
| | | <i>zmodload -e [string ...]</i> | Evaluates the given strings as Zsh code |