

## BASH QUICK REFERENCE CARD

© 2014 Jonas Petong - Permissions on back. v1.0 Mar 2014

### Basic Operations

`cd path`..... change directory  
    `..`          move one folder up  
    `-`          change to last visited directory  
    `<None>`      change to homedir (`~`)  
`mkdir fold`..... create new folder in current *path*

### Movements

for `<esc>` you can use whether `<esc>` or `<alt>`-key

`^a ^e`..... jump to beginning/end of the line  
`<esc f> / <esc b>`.. jump one word forward/backward  
`^c`..... cancel current command  
`^y`..... paste most recently cut out text  
`<esc d> / <esc ←>`. cut one word forward/backward  
`^w`..... cut text backward until space  
`^u / ^k`..... cut from current position to the beginning/end of line  
`^#`..... prepend hashtag on command, then jump to next line (... and reedit later)

### History

clearing your entire history event line: `history -c`

### Shortcuts

`^p`..... fetch the previous command from the history list  
`^r`..... search history backward (incremental search)  
`<esc ->`..... running last executed command  
`<esc p>`..... search backward using non-incremental search

`<esc n>`..... search forward using non-incremental search  
`<esc <>`..... move to the first line in the history  
`<esc >>`..... move to the end of the history list (current entry)

### History-Commands

example command is: `echo foo bar`

`!!`..... runs previous executed command (also known as ‘bang bang’)  
    `sudo !!`    `sudo echo foo bar`  
`!$ / <esc .>`..... repeats last *arg* of prev command  
`!n / !-n`..... executes command *n*/*n*th last command  
`!str / !?str?`..... runs last executed command, starting with/containing string or character *str* (ending `?` is optional)  
    `!echo`    `echo foo bar`  
    `!?` `sudo`    `sudo echo foo bar`  
`!n:d`..... executes first/last/specific/all *arg*’s of command *n* (refers to prev command, in case *n* is not given) → for meaning of designators *d*, see below!  
    `vim !sudo:*`    `vim echo foo bar`  
`!#`..... refers to the entire command line typed so far

### - Designators

0          the zeroth (first) word in a line (usually command name)  
*n*          the *n*th word in a line  
`^`          the first *arg* (the second word) in a line  
`$`          the last argument *arg* in a line  
`%`          the word *str* matched by the most recent `!?str?` search  
*m-n*      a range of words from *m* to *n* (*-n* is synonymous with `0-n`)

`*`          all *arg*’s  
*m*`*`        synonymous with *m-\$*  
*m-*        from *arg m* to end, omitting last word

### Substitutions & Replacements

`^str1^str2^`..... quick substitution. Repeats the last command, replacing *str1* with *str2*  
*mods*/old/new/  
                  substitutes new for old (for meaning of *mod* see below)

### - Modifiers (*mod*)

`h`          removes a trailing pathname component, leaving the head  
`t`          removes all leading pathname components, leaving the tail (filename in path)  
`r`          removes a trailing suffix of the form `.xxx`, leaving the basename  
`e`          removes all but the trailing suffix  
`p`          prints the resulting command but does not execute it  
`q`          quotes the substituted words, escaping further substitutions  
`m`          quotes the substituted words, breaking them into words at blanks and newlines  
`&`          repeats the previous substitution  
`g`          causes `s/old/new/` or `&` to be applied over the entire event line  
`G`          apply the following ‘s’ modifier just once to each word in the event line.

### Variables

append `export <setting>=<value>` to your `~/.bashrc`

`HISTFILE`..... if set to `/dev/null` history file is rejected (Default: `~/.bash_history`)

`HISTFILESIZE`

controls how many history commands to keep in `HISTFILE` (Default: 500)

HISTSIZE..... controls how many history commands to keep in the history list of current session (Default: 500)

## HISTCONTROL

modality of how to save commands on the history list.

ignorespace     if the list of values includes ignorespace, lines which begin with a space character are not saved in the history list.

ignoredups     a value of ignoredups causes lines matching the previous history entry to not be saved.

ignoreboth     a value of ignoreboth is shorthand for ignorespace and ignoredups.

erasedups     a value of erasedups causes all previous lines matching the current line to be removed from the history list before that line is saved.

## HISTIGNORE

controls which commands to ignore and not save to the history list.

## Shopt Option

shopt options can be set by a **shopt -s option** and can be unset by a **shopt -u option** shell command.

histappend..... setting the variable appends current session history to HISTFILE. Unsetting overwrites the file each time

histreedit..... if set, puts a failed history substitution back on the command line for re-editing

histverify..... if set, puts the command to be executed after a substitution on command line as if you had typed it