

## BASH QUICK REFERENCE CARD

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

### Shortcuts

`^p.....` Fetch the previous command from the history list

`^r.....` Search history backward (incremental search)

### History-Commands

example commmand is: `echo foo bar`

`!!.....` runs previous executed command (also known as ‘bang bang’)

`sudo !!` `sudo echo foo`

`!n !-n.....` executes command `n/nth` last command

`!c !?c?.....` runs last executed command, starting with/containing string or character `c` (ending `?` is optional)

`!echo` `echo foo bar`

`!? sudo` `sudo echo foo`

`!:d.....` executes first/last/nth/all arguments `arg` of prev command (for meaning of designators (`d`) see below!)

`vim !sudo:*` `vim echo`

`!#.....` Refers to the entire command line typed so far

### Substitutions & Replacements

`^c1^c2^.....` Quick substitution. Repeats the last command, replacing `c1` with `c2`

`mods/old/new/`

Substitutes new for old (for meaning of modifiers (`mod`) see below)

#### - 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 matched by the most recent `!?c?` 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

#### - Modifiers

`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 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

v1.0 Sep 2014. Copyright © 2014 Jonas Petong

Permission is granted to make and distribute copies of this card provided the copyright notice and this permission notice are preserved on all copies. Send comments or corrections to Jonas Petong at: [kaktuus@gmx.de](mailto:kaktuus@gmx.de)

<https://github.com/emzap79/latex/blob/master/qrcBash.tex>  
This TeXfile is based on Gabriel B. Burcas git-qrc.tex