

# Bash History Cheat Sheet

Emacs and Vi history editing keyboard shortcuts:

Shortcut	Description
<i>Emacs Mode Shortcuts:</i>	
<b>CTRL-p</b>	Fetch the previous command from the history list.
<b>CTRL-n</b>	Fetch the next command from the history list.
<b>CTRL-r</b>	Search history backward (incremental search).
<b>CTRL-s</b>	Search history forward (incremental search).
<b>Meta-p</b>	Search backward using non-incremental search.
<b>Meta-n</b>	Search forward using non-incremental search.
<b>Meta-&lt;</b>	Move to the first line in the history.
<b>Meta-&gt;</b>	Move to the end of the history list.
<i>Vi Mode Shortcuts:</i>	
<b>k</b>	Fetch the previous command from the history list.
<b>j</b>	Fetch the next command from the history list.
<b>/string</b> or <b>CTRL-r</b>	Search history backward for a command matching <i>string</i> .
<b>?string</b> or <b>CTRL-s</b>	Search history forward for a command matching <i>string</i> .
<b>n</b>	Repeat search in the same direction as previous.
<b>N</b>	Repeat search in the opposite direction as previous.
<b>G</b>	Move to the N-th history line (for example, <b>15G</b> ).

History behavior modification via shell variables:

Shell Variable	Description
<b>HISTFILE</b>	Controls where the history file gets saved. Set to <b>/dev/null</b> not to keep history. Default: <code>~/.bash_history</code> .
<b>HISTFILESIZE</b>	Controls how many history commands to keep in <b>HISTFILE</b> . Default: 500.
<b>HISTSIZE</b>	Controls how many history commands to keep in the history list of current session. Default: 500.
<b>HISTIGNORE</b>	Controls which commands to ignore and not save to the history list. The variable takes a list of colon separated patterns. Pattern <b>&amp;</b> matches the previous history command.

History behavior modification via *shopt* command:

shopt option	Description
<b>histappend</b>	Setting the variable appends current session history to <b>HISTFILE</b> . Unsetting overwrites the file each time.
<b>histreedit</b>	If set, puts a failed history substitution back on the command line for re-editing.
<b>histverify</b>	If set, puts the command to be executed after a substitution on command line as if you had typed it.

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

A cheat sheet by **Peteris Krumins** (peter@catonmat.net), 2008.  
<http://www.catonmat.net> - good coders code, great reuse

Released under GNU Free Document License.

## History expansion:

Shortcut	Description
<i>Event Designators:</i>	
!	Starts a history substitution.
!!	Refers to the last command.
! <b>n</b>	Refers to the <b>n</b> -th command line.
! <b>-n</b>	Refers to the current command line minus <b>n</b> .
! <b>string</b>	Refers to the most recent command starting with <b>string</b> .
! <b>?string?</b>	Refers to the most recent command containing <b>string</b> (the ending ? is optional).
^ <b>string1</b> ^ <b>string2</b> ^	Quick substitution. Repeats the last command, replacing <b>string1</b> with <b>string2</b> .
! <b>#</b>	Refers to the entire command line typed so far.
<i>Word Designators (word designators follow the event designators, separated by a colon):</i>	
<b>0</b>	The zeroth (first) word in a line (usually command name).
<b>n</b>	The <b>n</b> -th word in a line.
<b>^</b>	The first argument (the second word) in a line.
<b>\$</b>	The last argument in a line.
<b>%</b>	The word matched by the most recent ?string? search.
<b>x-y</b>	A range of words from <b>x</b> to <b>y</b> ( <b>-y</b> is synonymous with <b>0-y</b> ).
<b>*</b>	All word but the zeroth.
<b>x*</b>	Synonymous with <b>x-\$</b> .
<b>x-</b>	The words from <b>x</b> to the second to last word.
<i>Modifiers (modifiers follow word designators, separated by a colon):</i>	
<b>h</b>	Removes a trailing pathname component, leaving the head.
<b>t</b>	Removes all leading pathname components, leaving the tail.
<b>r</b>	Removes a trailing suffix of the form .xxx, leaving the basename.
<b>e</b>	Removes all but the trailing suffix.
<b>p</b>	Prints the resulting command but does not execute it.
<b>q</b>	Quotes the substituted words, escaping further substitutions.
<b>x</b>	Quotes the substituted words, breaking them into words at blanks and newlines.
<b>s/old/new/</b>	Substitutes <b>new</b> for <b>old</b> .
<b>&amp;</b>	Repeats the previous substitution.
<b>g</b>	Causes <b>s/old/new/</b> or <b>&amp;</b> to be applied over the entire event line.

## History expansion examples:

```
$ echo a b c d e      (executes 'echo ab c d e')
a b c d e
$ echo !!:3-$        (executes 'echo c d e')
c d e
$ echo !-2:*:q       (executes 'echo 'a b c d e'')
a b c d e
$ echo !-3:1:2:4:x   (executes 'echo 'a' 'b' 'd'')
a b d
$ echo !-4:1-3:s/a/foo/:s/b/bar/:s/c/baz/
(executes 'echo foo bar baz')
foo bar baz
```

```
$ tar -xzf package-x.y.z.tgz
...
$ cd !-1:$:r         (exec's 'cd package-x.y.z')
package-x.y.z $

$ ls -a /tmp
file1 file2 file3 ...
$ ^-a^-l^           (exec's 'ls -l /tmp')
-rw----- 1 user user file1
...
```