SHELL QUICK REFERENCE CARD

Movements

$\langle M \rangle$ denotes the meta-key, usually bound to $\langle alt \rangle$ _or_ $\langle esc \rangle$

`ˆa ˆe` . . . . jump to begin, end of the line

`ˆu ˆk` . . . . cut to beginning, end of the line

`ˆf ˆb` . . . . move one _char_ forward, backward

$\langle m \rangle$`f` $\langle m \rangle$`b` . . . jump one word forward, backward

$\langle m \rangle \leftarrow$ _or_ `ˆw` . . cut text backward until space

$\langle m \rangle$`d` . . . . cut one word forward until space

`ˆy` . . . . paste most recently cut out text

History

Prompt all executed history events: `history`, clear entire history event cache: `history -c`

Shortcuts

`ˆp` . . . . fetch the previous command from the history list

$\langle m \rangle$`.` . . . . print last argument(s) to line

`ˆr ˆs` . . . . search history backward, forward (incremental search)

$\langle m \rangle$`p` $\langle m \rangle$`n` . . . search backward, forward using non-incremental search

$\langle m \rangle<$ $\langle m \rangle>$ . . move to the first, last line in history

History-Commands

Example command is: `cd foo/bar`

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

$\Rightarrow$ `sudo !!:` `sudo cd foo/bar`

`!$` . . . . repeats last _arg_ of prev command

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

`!str !?str?` . . runs last executed command starting with, containing _str_

$\Rightarrow$ `!cd:` `cd foo/bar`

`!!:opt` . . . . filter last command by designators _or_ modifiers

$\Rightarrow$ `echo !cd:q:` `echo 'cd foo/bar'`

`!#` . . . . entire command line typed so far

`!!:gs/old/new/`
    substitutes new for old

$\Rightarrow$ quick substitution, replacing str_1 with str_2: `ˆstr₁ˆstr₂` ($\Rightarrow$ `ˆ` literally)

Options

you may append as many options as you want, e. g. you could do a: `!! : $ : p` (prints least argument of last executed history command)

Event Designators

0	the first word in a line (usually command _or_ programs name)
n	the nth word in a line
ˆ	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, ommitting last word

Event 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 just once to each word in the event line.

Special Commands

`ˆs ˆc ˆq` . . . . . pause, cancel, resume current process

`popd` . . . . switch to last saved _dir_ from index

`pushd dir` . . . . save _dir_ to index

`ˆXˆE` . . . . work on line under cursor in default EDITOR and execute it afterwards

`ˆ#` . . . . prepend hash to command, then jump to next line (... and reedit later)