



Faversham Associates Ltd

Bash

Command Line Cheat Sheet

Version: 1.0.0

Gavin Bungay

gavin.bungay@favershamassociates.com

December 10, 2017

Shortcut	Description
Ctrl-a	Move to the start of the line.
Ctrl-e	Move to the end of the line.
Alt-] x	Moves the cursor forward to the next occurrence of x.
Alt-Ctrl-] x	Moves the cursor backwards to the previous occurrence of x.
Ctrl-u	Delete from the cursor to the beginning of the line.
Ctrl-k	Delete from the cursor to the end of the line.
Ctrl-w	Delete from the cursor to the start of the word.
Ctrl-y	Pastes text from the clipboard.
Ctrl-l	Clear the screen leaving the current line at the top of the screen.
Ctrl-x Ctrl-u	Undo the last changes. Ctrl-_
Alt-r	Undo all changes to the line.
Alt-Ctrl-e	Expand command line.
Ctrl-r	Incremental reverse search of history.
Alt-p	Non-incremental reverse search of history.



Shortcut	Description
!!	Execute last command in history.
!*	Recalls all the parameters from the last command.
!!:n	Recalls the nth parameter from the last command.
!abc	Execute last command in history beginning with abc.
!n	Execute nth command in history.
^abc^xyz	Replace first occurrence of abc with xyz in last command and execute it.
cd	Change to user's home directory.
cd -	Change to previous directory.

Table 1: *Bash Command Line Shortcuts.*

Examples

- Renaming multiple files in a directory and featuring command line substitution:

```
$ for i in *cms*.tex; do mv $i $i/cms/CMS; done
```

(Renames files with 'cms' in their name to contain CMS instead.)

- Creating multiple subdirectories together with their full paths:

```
$ mkdir -p /rootdir/subdir1/subdir2/{dir1,dir2,dir3}
```

(Creates 3 directories under subdir2, the -p creates the full paths if they don't already exist.)

- Renaming/moving files with suffixes quickly:

```
$ cp /home/foo/realllylongname.cpp{,-old}
```

This expands to:

```
$ cp /home/foo/realllylongname.cpp /home/foo/realllylongname.cpp-old
```



- To split a file name into the base name and extension:

To get the basename:

```
$ echo ${filename%.*}
```

To get the extension:

```
$ echo ${filename#*.*}
```

To split all the files into a directory:

```
$ for i in *; do echo ${i%.*} ${i#*.*} ; done
```

- To read a directory listing into an array:

```
$ files=(`ls`), (note back tick);
```

To get n^{th} file:

```
$ echo ${files[n]}
```

- Replace and execute, use `^old^new`, eg:

```
$ vin <long path-file>
```

```
$ ^vin^vim
```

- List directories only:

```
$ ls -d */
```