

bash cheat sheet



Copyright (C) 2014
Felix C. Stegerman <flx@obfusk.net>

getting help

help <builtin>
man <command>

history

history
!! !foo

special characters

~`#\$%*()\[\]{}; '"/?!

options

set shopt

quoting

echo 'single-quoted: "hi!" ...'
echo "double-quoted: '\\$foo=\$foo'"

streams

cat	head	tail	
awk	grep	sed	tr
rev	shuf	sort	uniq
seq	wc		
column	cut	paste	tee

miscellaneous

id	groups	who
mktemp	renice	
printf	iconv	strings
crontab	time	watch
tar	zip	
whiptail	zenity	
perl	python	ruby
less	nano	vim
ack	git	tree
diff	patch	
cal	date	
mutt	rsync	screen

redirecting I/O

command > file
command >> append
command 2> errors
command < input-file
command >&2 # to stderr
command <<<"this is input"
command >/dev/null # bye

files and directories

ls -hl --color=auto

cd	pwd	pushd	popd
cp	ln	mv	rm
mkdir	rmdir		rename
chown	chmod	touch	umask
du	file	stat	test
readlink		which	
basename		dirname	

command substitution

cmd1 "\$(cmd2 ...)"
diff <(cmd1) <(cmd2)

shell scripts

#!/bin/bash
set -e
...

arithmetic

((i = 99, j = i * i)) bc
echo \$((i == 99 ? 42 : 37))

printing

lpr lpq lprm lpstat

networking

hostname
ifconfig
ping
curl wget
nc

key bindings

^C SIGTERM	^D EOF
^A home	^E end
TAB complete	^L clear
	UP/DOWN history

processes & signals

in-background &	
bg	fg jobs ^Z
sleep	wait kill ps
nohup	trap

exit codes (\$?)

0 success >0 failure

globbing

ls ?[^a-z]*
mv foo-{bar,baz}

system

free top
su sudo

arrays

my_ary=(foo bar "a b c")
echo "\${my_ary[2]}"
command "\${my_ary[@]}"

find

find /foo /bar -type d \
-print0 | sort -z | \
xargs -0 rmdir

variables

foo="some value"

echo \$foo
echo "\$foo"
echo "\${foo:0:4}" # "some"
echo \${foo:+alternative}
echo \${foo:-default}
: \${foo:=assigned_default}
echo \${#foo} # length
echo \${foo#prefix} # rm prefix
echo \${foo%.ext} # rm suffix
echo \${foo/some/replaced}

declare export local unset env

special variables

\$?	\$_	@	\$1	\$2	...
\$PATH	\$HOME	\$LINES	\$COLUMNS		
\$PS1			shift		

pipes

cmd1 | cmd2 | cmd3
echo \${PIPESTATUS[@]}
mkfifo a b; cmd1 <a >b; cmd2 a

locales

locale
\$LC_*

filesystems

mount fsck
df dd

scripting

if [-e existing_file]; then echo YES; else echo NO; fi
while read -r; do echo "[\$REPLY]"; done
for x in "\$@"; do echo "[\$x]"; done
for ((i=0; i<10; ++i)); do echo "\$i"; done
case foo in f*) echo Y;; *) echo N;; esac
select x in A B C; do echo \$x; [-z "\$x"] || break; done

[["\$x" =~ ^foo]]
cmd1 && cmd2 cmd1 || true !false
(echo "subshell") { cmd1; cmd2; cmd3; }

foo () { echo HI; } # function

eval "\$hopefully_not_evil_code"

source

getopts

exit
return