

# UNIX / Linux Shell Cheat Sheet



File Manipulation	
> file	create (overwrite) file
>> file	append to file
>file 2>&1	both output and errors to file
< file	read from file
a   b	pipe output from 'a' as input to 'b'

Common Constructs	
while read f do echo "Line is \$f" done < file	read text file line by line
\$ grep foo myfile afoo foo foobar	find matching lines
\$ cut -d: -f5 /etc/passwd Dilbert	get field with delimiter
foo=`ls`	get output of command
case \$foo in a) echo "foo is A" ;; b) echo "foo is B" ;; *) echo "foo is not A or B" ;; esac	case is a good way to avoid iterating through many if/elif/elif/elif constructs.
doubleit() { expr \$1 \* 2 } doubleit 3 # returns 6	function declaration and calling syntax
for i in * do echo "File is \$i" done	A for loop iterates through its input (which is subject to globbing)

Useful Variables	
\$IFS	Internal File Separator
\$?	return code from last program
\$SHELL	what shell is running this script?
LANG	Language; C is US English

Test Operators	
if [ "\$x" -lt "\$y" ]; then # do something fi	

Numeric Tests	
lt	less than
gt	greater than
eq	equal to
ne	not equal
ge	greater or equal
le	less or equal

File Tests	
nt	newer than
d	is a directory
f	is a file
r	readable
w	writable
x	executable

String Tests	
=	equal to
z	zero length
n	not zero length

Logical Tests	
&&	logical AND
	logical OR
!	logical NOT

Argument Variables	
\$0	program name
\$1	1 <sup>st</sup> argument
\$2	2 <sup>nd</sup> argument
...	...
\$9	9 <sup>th</sup> argument
\$*	all arguments
\$#	No. of arguments

Variable Substitution	
\${V:-def}	\$V, or "def" if unset
\${V:=def}	\$V (set to "def" if unset)
\${V:?err}	\$V, or "err" if unset

Conditional Execution	
c1    c2	run c1; if it fails, run c2
c1 && c2	run c1; if it works, run c2

Common utilities and switches	
ls -lSr	list files, biggest last
ls -ltr	list files, newest last
ls -lh	human-readable file sizes
du -sk *	directory sizes (slow)
sort -n	sort numerically (not alpha)
ps -ef	list my commands
wget URL	download URL
time cmd	stopwatch on `cmd`
touch file	create file
read x	read "x" from keyboard
cmd   \	tee file.txt
tee file.txt	cmd output to stdout and also to file.txt
nice cmd	run cmd with low priority

Networking	
ifconfig -a	list all network interfaces
netstat -r	show routers
ssh u@host	log in to host as user "u"
scp file.txt \	copy file.txt to host as
u@host:	user "u"

General Admin	
less file	display file, page by page
alias l='ls -l'	create "l" as alias for "ls -l"
tar cf t.tar \	create a tar archive t.tar
list_of_files	from the listed dirs/files
cal 3 1973	display a calendar (Mar 73)
df -h	show disk mounts
truss -p PID	show syscalls of PID

Files: Contents / Attributes	
find . -size 10k -print	files over 10Kb
find . -name "*.txt" -print	find text files
find /foo -type d -ls	ls all directories under /foo
three=`expr 1 + 2`	simple maths
echo "scale = 5 ; 5121 / 1024"   bc	better maths
egrep "(foo bar)" file	find "foo" or "bar" in file
awk '{ print \$5 }' file	print the 5 <sup>th</sup> word of each line
sed s/foo/bar/g file	replace "foo" in file with "bar"