

Shell Scripting Cheat Sheet
for Unix and LinuxOnline: <http://steve-parker.org/sh/sh.shtml>
Book: <http://steve-parker.org/shellscripting>

File Redirection

> file	create (overwrite) file
>> file	append to file
< file	read from file
a b	Pipe 'a' as input to 'b'

Common Constructs

\$ while read f	read text file
> do	line by line
> echo "Line is \$f"	
> done < file	note: "\$" prompt becomes ">"
\$ grep foo myfile	find lines in
afoo	myfile
foo	containing the
foobar	text "foo"
\$ cut -d: -f5 /etc/passwd	get 5 th field
Steve Parker	delimited by colon
\$ cmd1 cmd2	run cmd1; if fails, run cmd2
\$ cmd1 && cmd2	run cmd1; if it works, run cmd2
case \$foo in	act upon the
a)	value of a
echo "foo is A" ;;	variable
b)	
echo "foo is B" ;;	note that ";;"
*)	is required
echo "foo is not A or B"	at the end of
;;	each section
esac	
myvar=`ls`	get output of
	ls into variable
doubleit() {	function
expr \$1 * 2	declaration
}	and syntax
doubleit 3 # returns 6	for calling it

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
x	executable
r	readable
w	writable

String Tests

=	equal to
z	zero length
n	not zero length

Logical Tests

&&	logical AND
	logical OR
!	logical NOT

Arguments

\$0	program name
\$1	1 st argument
\$2	2 nd argument
...	...
\$#	no. of arguments
\$*	all arguments

Variable Substitution

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

Conditional Execution

cmd1 cmd2	run cmd1; if fails, run cmd2
cmd1 && cmd2	run cmd1; if ok, run cmd2

Files

mv /src /dest	move /src into /dest
ls a*	list files beginning with "a"
ls *a	list files ending with "a"
ls -ltr	list oldest first, newest last
ls -lsr	list smallest first, biggest last
ls -la	list all files, including hidden
find /src -print \	copy /src into current
cpio -pudvm	directory, preserving links, special devices, etc.

Preset Variables

\$SHELL	what shell am I running?
\$RANDOM	provides random numbers
\$\$	PID of current process
\$?	return code from last cmd
\$_	PID of last background cmd

Generally Useful Commands

file /etc/hosts	determine file type
basename /bin/ls	strip directory name (ls)
dirname /bin/ls	get directory name (/bin)
ifconfig -a	show all network adapters
netstat -r	show routers
netstat -a	show open ports
date +%Y%m%d	Year, Month, Day
date +%H%M	Hours, Minutes
wc -l	count number of lines
pwd	present working directory

Misc Useful Commands and Tools

egrep "(foo bar)" file	find "foo" or "bar" in file
awk '{ print \$5 }' file	print the 5 th word of each line
cal 3 1973	March 1973
df -h	show disk mounts
three=`expr 1 + 2`	simple maths
echo "scale = 5 ; \	better maths
5121 / 1024" bc	(5.00097)
time cmd	stopwatch on cmd
touch file	create blank file
alias ll='ls -l'	alias for ls -l
unalias ls	unset existing alias
find . -size 10k -print	files over 10Kb
find . -name "*.txt" -print	find text files
find /foo -type d -ls	list all directories under /foo
less file	display file page by page
sed s/foo/bar/g file	replace "foo" with "bar"
sed -i s/foo/bar/g file	in file (-i: update file)
strace -tfp PID	trace system calls for PID
tar cvf archive.tar file1 file 2 file3	create tar archive
ssh user@host	log in to host as user
scp file.txt user@host:	copy file.txt to host as user
scp user@host:/tmp/file.txt /var/tmp	copy /tmp/file.txt from user at host to /var/tmp locally
cd -	return to previous directory