

--help and help

Many of the commands on your system will generate a brief discussion of usage and options if you run them with the **--help** option. For example, trying this with **rm** by doing **rm --help** gives the output seen in the screenshot below.

```
File Edit View Search Terminal Help
c7:/tmp>rm --help
Usage: rm [OPTION]... FILE...
Remove (unlink) the FILE(s).

  -f, --force            ignore nonexistent files and arguments, never prompt
  -i                    prompt before every removal
  -I                    prompt once before removing more than three files, or
                        when removing recursively; less intrusive than -i,
                        while still giving protection against most mistakes
  --interactive[=WHEN]  prompt according to WHEN: never, once (-I), or
                        always (-i); without WHEN, prompt always
  --one-file-system     when removing a hierarchy recursively, skip any
                        directory that is on a file system different from
                        that of the corresponding command line argument
  --no-preserve-root    do not treat '/' specially
  --preserve-root       do not remove '/' (default)
  -r, -R, --recursive  remove directories and their contents recursively
  -d, --dir             remove empty directories
  -v, --verbose         explain what is being done
  --help               display this help and exit
  --version             output version information and exit

By default, rm does not remove directories.  Use the --recursive (-r or -R)
option to remove each listed directory, too, along with all of its contents.

To remove a file whose name starts with a '-', for example '-foo',
use one of these commands:
  rm -- -foo

  rm ./-foo

Note that if you use rm to remove a file, it might be possible to recover
some of its contents, given sufficient expertise and/or time.  For greater
assurance that the contents are truly unrecoverable, consider using shred.

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
For complete documentation, run: info coreutils 'rm invocation'
c7:/tmp>
```

This is often all you need and can be consumed much quicker than running **man** or **info**.

There is also a **help** command, which is actually part of the bash shell, and only gives information about commands which are actually part of the shell itself. Typing **help** by itself generates the screenshot shown below,

File Edit View Search Terminal Help

```
c7:/home/coop>help
GNU bash, version 4.2.46(1)-release (x86_64-redhat-linux-gnu)
These shell commands are defined internally. Type 'help' to see this list.
Type 'help name' to find out more about the function 'name'.
Use 'info bash' to find out more about the shell in general.
Use 'man -k' or 'info' to find out more about commands not in this list.

A star (*) next to a name means that the command is disabled.

job_spec [&]
(( expression ))
. filename [arguments]
:
[ arg... ]
[[ expression ]]
alias [-p] [name=value] ... ]
bg [job_spec ...]
bind [-lpvsPVS] [-m keymap] [-f filename] [-q name] [>
break [n]
builtin [shell-builtin [arg ...]]
caller [expr]
case WORD in [PATTERN] [PATTERN]... ) COMMANDS ;;...>
cd [-L|[-P [-e]]] [dir]
command [-pVv] command [arg ...]
compgen [-abcefgjksuv] [-o option] [-A action] [-G >
complete [-abcefgjksuv] [-pr] [-DE] [-o option] [-A >
compopt [-o|+o option] [-DE] [name ...]
continue [n]
coproc [NAME] command [redirections]
declare [-aAfFgilrtux] [-p] [name=value] ...]
dirs [-clpv] [+N] [-N]
disown [-h] [-ar] [jobspec ...]
echo [-neE] [arg ...]
enable [-a] [-dnps] [-f filename] [name ...]
eval [arg ...]
exec [-cl] [-a name] [command [arguments ...]] [redir>
exit [n]
export [-fn] [name=value] ...] or export -p
false
fc [-e ename] [-lnr] [first] [last] or fc -s [pat=rep>
fg [job_spec]
for NAME [in WORDS ... ] ; do COMMANDS; done
for (( exp1; exp2; exp3 )); do COMMANDS; done
function name { COMMANDS ; } or name () { COMMANDS ; >
getopts optstring name [arg]
hash [-lr] [-p pathname] [-dt] [name ...]
help [-dms] [pattern ...]
history [-c] [-d offset] [n] or history -anrw [file>
if COMMANDS; then COMMANDS; [ elif COMMANDS; then CO>
jobs [-lnprs] [jobspec ...] or jobs -x command [args>
kill [-s sigspec | -n signum | -sigspec] pid | jobsp>
let arg [arg ...]
local [option] name[=value] ...
logout [n]
mapfile [-n count] [-O origin] [-s count] [-t] [-u f>
popd [-n] [+N | -N]
printf [-v var] format [arguments]
pushd [-n] [+N | -N | dir]
pwd [-LP]
read [-ers] [-a array] [-d delim] [-i text] [-n ncha>
readarray [-n count] [-O origin] [-s count] [-t] [-u>
readonly [-aAf] [name=value] ...] or readonly -p
return [n]
select NAME [in WORDS ... ] do COMMANDS; done
set [-abefghkmnpstuvxBCHP] [-o option-name] [--] [arg >
shift [n]
shopt [-pqsu] [-o] [optname ...]
source filename [arguments]
suspend [-f]
test [expr]
time [-p] pipeline
times
trap [-lp] [[arg] signal_spec ...]
true
type [-afptP] name [name ...]
typeset [-aAfFgilrtux] [-p] name[=value] ...
ulimit [-SHacdefilmnpqrstuvx] [limit]
umask [-p] [-S] [mode]
unalias [-a] name [name ...]
unset [-f] [-v] [name ...]
until COMMANDS; do COMMANDS; done
variables - Names and meanings of some shell variabl>
wait [id]
while COMMANDS; do COMMANDS; done
{ COMMANDS ; }
```

and information on a particular command can be done as in:

```
1 $ help pwd
2
3 pwd: pwd [-LP]
4     Print the current working directory. With the -P option, pwd prints
5     the physical directory, without any symbolic links; the -L option
6     makes pwd follow symbolic links.
```

It is important to note that there are programs which have two incarnations, one in the bash shell and one as a standalone program. For example, these two commands are similar but not identical:

```
1 $ echo hello
2 $ /bin/echo hello
```

By default, the command built into the shell is invoked, rather than the one in the path. Likewise, the results of **man echo** and **help echo** are not the same. This can be confusing.