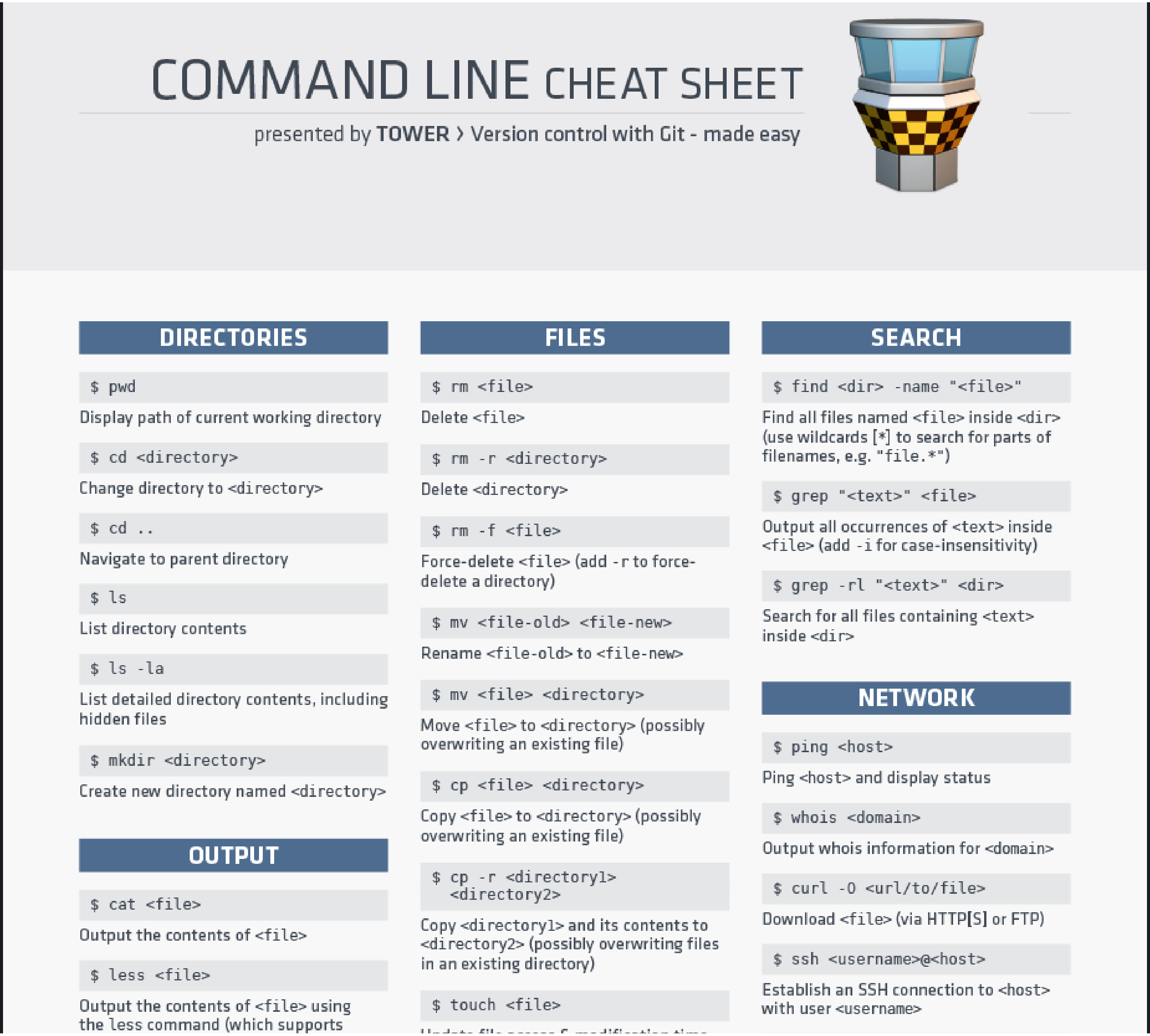
January 2017 by Tobias Günther

Command Line Cheat Sheet

Our cheat sheet explains the essential tasks on the command line. Download it for free.

For many, the command line belongs to long gone days: when computers were controlled by typing mystical commands into a black window; when the mouse possessed no power. But for many use cases, the command line is still absolutely indispensable!
Our cheat sheet not only features the most important commands. On the back, it also explains some tips & tricks that make working with the CLI a lot easier.

fournova



COMMAND LINE CHEAT SHEET

presented by **TOWER** › Version control with Git - made easy

DIRECTORIES

```
$ pwd
```
Display path of current working directory

```
$ cd <directory>
```
Change directory to <directory>

```
$ cd ..
```
Navigate to parent directory

```
$ ls
```
List directory contents

```
$ ls -la
```
List detailed directory contents, including hidden files

```
$ mkdir <directory>
```
Create new directory named <directory>

OUTPUT

```
$ cat <file>
```
Output the contents of <file>

```
$ less <file>
```
Output the contents of <file> using the less command (which supports

FILES

```
$ rm <file>
```
Delete <file>

```
$ rm -r <directory>
```
Delete <directory>

```
$ rm -f <file>
```
Force-delete <file> (add -r to force-delete a directory)

```
$ mv <file-old> <file-new>
```
Rename <file-old> to <file-new>

```
$ mv <file> <directory>
```
Move <file> to <directory> (possibly overwriting an existing file)

```
$ cp <file> <directory>
```
Copy <file> to <directory> (possibly overwriting an existing file)

```
$ cp -r <directory1>
  <directory2>
```
Copy <directory1> and its contents to <directory2> (possibly overwriting files in an existing directory)

```
$ touch <file>
```

SEARCH

```
$ find <dir> -name "<file>"
```
Find all files named <file> inside <dir> (use wildcards [*] to search for parts of filenames, e.g. "file.*")

```
$ grep "<text>" <file>
```
Output all occurrences of <text> inside <file> (add -i for case-insensitivity)

```
$ grep -rl "<text>" <dir>
```
Search for all files containing <text> inside <dir>

NETWORK

```
$ ping <host>
```
Ping <host> and display status

```
$ whois <domain>
```
Output whois information for <domain>

```
$ curl -O <url/to/file>
```
Download <file> (via HTTP[S] or FTP)

```
$ ssh <username>@<host>
```
Establish an SSH connection to <host> with user <username>

pagination etc.)

```
$ head <file>
```

Output the first 10 lines of <file>

```
$ <cmd> > <file>
```

Direct the output of <cmd> into <file>

```
$ <cmd> >> <file>
```

Append the output of <cmd> to <file>

```
$ <cmd1> | <cmd2>
```

Direct the output of <cmd1> to <cmd2>

```
$ clear
```

Clear the command line window

Update file access & modification time (and create <file> if it doesn't exist)

PERMISSIONS

```
$ chmod 755 <file>
```

Change permissions of <file> to 755

```
$ chmod -R 600 <directory>
```

Change permissions of <directory> (and its contents) to 600

```
$ chown <user>:<group> <file>
```

Change ownership of <file> to <user> and <group> (add -R to include a directory's contents)

```
$ scp <file>
  <user>@<host>:/remote/path
```

Copy <file> to a remote <host>

PROCESSES

```
$ ps ax
```

Output currently running processes

```
$ top
```

Display live information about currently running processes

```
$ kill <pid>
```

Quit process with ID <pid>

fournova

COMMAND LINE TIPS & TRICKS

presented by **TOWER** › Version control with Git - made easy

GETTING HELP

On the command line, help is always at hand: you can either type `man <command>`

THE "CTRL" KEY

Various keyboard shortcuts can assist you when entering text: Hitting `CTRL+A`

HOME FOLDER

File and directory paths can get long and awkward. If you're addressing a or `<command> --help` to receive detailed documentation about the command in question.

FILE PERMISSIONS

On Unix systems, file permissions are set using three digits: the first one representing the permissions for the owning user, the second one for its group, and the third one for anyone else.

Add up the desired access rights for each digit as following:

`4` – access/read (r)
`2` – modify/write (w)
`1` – execute (x)

For example, `755` means “rwx” for owner and “rx” for both group and anyone. `740` represents “rwx” for owner, “r” for group and no rights for other users.

COMBINING COMMANDS

If you plan to run a series of commands after another, it might be useful to combine them instead of waiting for each command to finish before typing the next one. To do so, simply separate the commands with a semicolon (`;`) on the same line.

Additionally, it is possble to execute a command only if its predecessor produces a certain result. Code placed after the `&&` operator will only be run if the previous command completes successfully, while the opposite `||` operator only continues if the previous command fails. The following command will create the folder “videos” only if the `cd` command fails (and the folder therefore doesn't exist):

moves the caret to the beginning and `CTRL+E` to the end of the line.

In a similar fashion, `CTRL+K` deletes all characters after and `CTRL+U` all characters in front of the caret.

Pressing `CTRL+L` clears the screen (similarly to the `clear` command). If you should ever want to abort a running command, `CTRL+C` will cancel it.

THE “TAB” KEY

Whenever entering paths and file names, the `TAB` key comes in very handy. It autocompletes what you've written, reducing typos quite efficiently. E.g. when you want to switch to a different directory, you can either type every component of the path by hand:

```
$ cd ~/projects/acmedesign/docs/
```

...or use the `TAB` key (try this yourself):

```
$ cd ~/pr[TAB]ojects/
  ac[TAB]medesign/d[TAB]ocs/
```

In case your typed characters are ambiguous (because “ac” could point to the “acmedesign” or the “actionscript” folder), the command line won't be able to autocomplete. In that case, you can hit `TAB` twice to view all possible matches and then type a few more characters.

THE ARROW KEYS

The command line keeps a history of the most recent commands you executed. By pressing the `ARROW UP` key, you can step through the last called commands (starting with the most recent). `ARROW DOWN` will move forward in history towards the most recent call.

Bonus tip: Calling the `history` command

path inside of your home folder though, you can make things easier by using the `~` character. So instead of writing `cd /Users/your-username/projects/`, a simple `cd ~/projects/` will do.

And in case you should forget your user name, `whoami` will remind you.

OUTPUT WITH “LESS”

The `less` command can display *and paginate* output. This means that it only displays one page full of content and then waits for your explicit instructions. You'll know you have `less` in front of you if the last line of your screen either shows the file's name or just a colon (`:`).

Apart from the arrow keys, hitting `SPACE` will scroll one page forward, `b` will scroll one page backward, and `q` will quit the `less` program.

DIRECTING OUTPUT

The output of a command does not necessarily have to be printed to the command line. Instead, you can decide to direct it to somewhere else.

Using the `>` operator, for example, output can be directed to a file. The following command will save the running processes to a text file in your home folder:

```
$ ps ax > ~/processes.txt
```

It is also possible to pass output to another command using the `|` (pipe) operator, which makes it very easy to create complex operations. E.g., this chain of commands will list the current directory's contents, search the list for PDF files and display the results with the `less` command:

```
$ cd ~/videos || mkdir ~/videos
```

Bonus tip: calling the `history` command prints a list of all recent commands.

```
$ ls | grep ".pdf" | less
```

30-day free trial available at www.git-tower.com

TOWER
Version control with Git - made easy

Download the Cheat Sheet

Sign up for our free "Cheat Sheet Package" - and get 6 of our most popular Cheat Sheets right to your inbox!

Your email address

We won't send you spam. Unsubscribe at any time.

We make Tower

...the best Git client for Mac and Windows.

We help over 100,000 users in companies like Apple, Google, Amazon, Twitter, and Facebook to easily & productively work with the Git version control system.

Try it 30 days for free!

Read next

...popular & related articles.

- › Tower in Public Beta: Here's What's New!
- › Git Cheat Sheet
- › Diff Tools on macOS