Command Line Cheat Sheet

DIRECTORIES

Display path of current working directory

```
$ pwd
```

Change directory to <directory>

```
$ cd <directory>
```

Navigate to parent directory

```
$ cd ..
```

List directory contents

```
$ ls
```

List detailed directory contents, including hidden files

```
$ ls -la
```

Create new directory named <directory>

```
$ mkdir <directory>
```

FILES

Delete <file>

```
$ rm <file>
```

Force-delete <file> (add -r to force-delete a directory)

```
$ rm -f <file>
```

Delete <directory>

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

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

```
$ cp <file> <directory>
```

Copy <directory1> and its contents to <directory2> (possibly overwriting files in an existing directory)

```
$ cp -r <directory1> <directory2>
```

Rename <file-old> to <file-new>

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

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

```
$ mv <file> <directory>
```

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

```
$ touch <file>
```

PERMISSIONS

Change permissions of <file> to 755

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

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

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

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

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

SEARCH

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

```
$ find <dir> -name "<file>"
```

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

```
$ grep "<text>" <file>
```

Search for all files containing <text> inside <dir>

```
$ grep -rl "<text>" <dir>
```

OUTPUT

Output the contents of <file>

```
$ cat <file>
```

Output the contents of <file> using the less command (which supports pagination etc.)

```
$ less <file>
```

Output the first 10 lines of <file>

```
$ head <file>
```

Direct the output of <cmd> into <file>

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

Append the output of <cmd> to <file>

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

Direct the output of <cmd1> to <cmd2>

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

Clear the command line window

```
$ clear
```

NETWORK

Ping <host> and display status

```
$ ping <host>
```

Output whois information for <domain>

```
$ whois <domain>
```

Download <file> (via HTTP[S] or FTP)

```
$ curl -O <url/to/file>
```

Establish an SSH connection to <host> with user <username>

```
$ ssh <username>@<host>
```

Copy <file> to a remote <host>

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

PROCESSES

Output currently running processes

```
$ ps ax
```

Display live information about currently running processes

```
$ top
```

Quit process with ID <pid>

```
$ kill <pid>
```

GETTING HELP

Get detailed documentation about a command

```
$ man <command>
```

Get help for a command

```
$ <command> --help
```

FILE PERMISSIONS

On Unix systems, file permissions are set using three digits:

- First digit: permissions for the owning user
- Second digit: permissions for the group
- Third digit: permissions for anyone else

Add up the desired access rights for each digit:

- access/read (r) = 4
- modify/write (w) = 2
- execute (x) = 1

Examples:

- 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

Run commands in sequence (separated by semicolon)

```
$ command1; command2; command3
```

Run command2 only if command1 succeeds

```
$ command1 && command2
```

Run command2 only if command1 fails

```
$ command1 || command2
```

Example: Create folder "videos" only if cd command fails

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

KEYBOARD SHORTCUTS

CTRL Key Shortcuts:

- `CTRL+A` - Move cursor to beginning of line
- `CTRL+E` - Move cursor to end of line
- `CTRL+K` - Delete all characters after cursor
- `CTRL+U` - Delete all characters before cursor
- `CTRL+L` - Clear the screen (same as `clear` command)
- `CTRL+C` - Cancel/abort running command

TAB Key:

- Auto-complete paths and file names
- Hit TAB twice to view all possible matches if ambiguous

Arrow Keys:

- ARROW UP - Step through previous commands (most recent first)
- ARROW DOWN - Move forward in command history

History Command:

```
$ history
```

HOME FOLDER

Use ~ character to reference home folder

```
$ cd ~/projects/
```

instead of

```
$ cd /Users/your-username/projects/
```

Show current username

```
$ whoami
```

OUTPUT WITH "LESS"

Navigation in less:

- SPACE - Scroll one page forward
- b - Scroll one page backward
- q - Quit less program
- Arrow keys - Navigate line by line

DIRECTING OUTPUT

Save command output to file

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

Pipe output to another command

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

This example lists directory contents, searches for PDF files, and displays results with less.