UNIX Cheat Sheet

(adapted from Treebeard's Unix Cheat Sheet, http://www.rain.org/~mkummel/unix.html)

Help on any Unix command.

`man {command}`	Type **man ls** to read the manual for the **ls** command.
`man {command} > {filename}`	Redirect help to a file to download.

List a directory

`ls {path}`	It's ok to combine attributes, eg **ls -laF** gets a long listing of all files with types.
`ls {path_1} {path_2}`	List both {path_1} and {path_2}.
`ls -l {path}`	Long listing, with date, size and permisions.
`ls -a {path}`	Show all files, including important .dot files that don't otherwise show.
`ls -F {path}`	Show type of each file. "/" = directory, "*" = executable.
`ls -R {path}`	Recursive listing, with all subdirs.
`ls {path} > {filename}`	Redirect directory to a file.
`ls {path} | more`	Show listing one screen at a time.

Change to directory

`cd {dirname}`	There must be a space between.
`cd ~`	Go back to home directory, useful if you're lost.
`cd ..`	Go back one directory.

Make a new directory

```
mkdir {dirname}
```

Remove a directory

`rmdir {dirname}`	Only works if {dirname} is empty.
`rm -r {dirname}`	Remove all files and subdirs. Careful!

Print working directory

`pwd`	Show where you are as full path. Useful if you're lost or exploring.

Change user password

`passwd`	Change user password

Copy a file or directory

`cp {file1} {file2}`	
`cp -r {dir1} {dir2}`	Recursive, copy directory and all subdirs.
`cat {newfile} >> {oldfile}`	Append newfile to end of oldfile.

Move (or rename) a file

`mv {oldfile} {newfile}`	Moving a file and renaming it are the same thing.
`mv {oldname} {newname}`	

Delete a file

`rm {filespec}`	**?** and ***** wildcards: "?" is any character; "*" is any string of characters.
`ls {filespec}`	Good strategy: first list a group to make sure it's what's you think...
`rm {filespec}`	...then delete it all at once.

View a text file

`more {filename}`	View file one screen at a time.
`less {filename}`	Like **more**, with extra features.
`cat {filename}`	View file, but it scrolls.
`cat {filename} | more`	View file one screen at a time.

Create and edit a text file.

```
emacs {filename}
pico {filename}
vi {filename}
```

Compare two files

`diff {file1} {file2}`	Show the differences.
`sdiff {file1} {file2}`	Show files side by side.

Other text commands

`grep '{pattern}' {file}`	Find regular expression in file.
`sort {file1} > {file2}`	Sort file1 and save as file2.
`sort -o {file} {file}`	Replace file with sorted version.
`spell {file}`	Display misspelled words.
`Wc {file}`	Count words in file.

Find files on system

`find {filespec}`	Works with wildcards. Handy for snooping.
`find {filespec} > {filename}`	Redirect find list to file. Can be big!

Make an Alias

`alias {name} '{command}'`	Put the command in 'single quotes'. More useful in your **.cshrc** file.

Wildcards and Shortcuts

`*`	Match any string of characters, eg **page*** gets page1, page10, and page.txt.
`?`	Match any single character, eg **page?** gets page1 and page2, but not page10.
`[...]`	Match any characters in a range, eg **page[1-3]** gets page1, page2, and page3.
`~`	Short for your home directory, eg **cd** ~ will take you home, and **rm -r** ~ will destroy it.
`.`	The current directory.
`..`	One directory up the tree, eg **ls ..**.

Pipes and Redirection

(You **pipe** a command to another command, and **redirect** it to a file.)

`{command} > {file}`	Redirect output to a file, eg **ls > list.txt** writes directory to file.
`{command} >> {file}`	Append output to an existing file, eg **cat update >> archive** adds update to end of archive.
`{command} < {file}`	Get input from a file, eg **sort < file.txt**
`{command} < {file1} > {file2}`	Get input from `file1`, and write to `file2`, eg **sort < old.txt > new.txt** sorts old.txt and saves as new.txt.
`{command} | {command}`	Pipe one command to another, eg **ls | more** gets directory and sends it to **more** to show it one page at a time.

Permissions, important and tricky!

Unix permissions concern who can **read** a file or directory, **write** to it, and **execute** it.

You can change file permissions with letters:

u = user (yourself)	**g** = group	**a** = everyone
r = read	**w** = write	**x** = execute

`chmod u+rw {filespec}`	Give yourself read and write permission
`chmod u+x {filespec}`	Give yourself execute permission.
`chmod a+rw {filespec}`	Give read and write permission to everyone.

System info

`date`	Show date and time.
`df`	Check system disk capacity.
`du`	Check your disk usage and show bytes in each directory.

Unix Directory Format

Long listings (**ls -l**) have this format:

```
- file
d directory,                                               * executable
^   symbolic links (?)  file size (bytes)   file name    / directory
^           ^               ^                  ^         ^
drwxr-xr-x 11 mkummel      2560 Mar  7 23:25 public_html/
-rw-r--r--  1 mkummel     10297 Mar  8 23:42 index.html
                                        ^
 ^^^        user permission  (rwx)      date and time last modified
    ^^^     group permission (rwx)
       ^^^  world permission (rwx)
```

Dotfiles (aka Hidden Files)

Dotfile names begin with a "." These files and directories don't show up when you list a directory unless you use the **-a** option, so they are also called **hidden files**. Type **ls -la** in your home directory to see what you have.

Some of these dotfiles are crucial. They initialize your shell and the programs you use. **rc** means "run commands". These are all text files that can be edited, but change them at your peril. Make backups first!

Here's some of what I get when I type **ls -laF**:

.cshrc	my C-shell startup info, important!
.history	list of past commands.
.login	login init, important!