Basic UNIX Cheat Sheet

Produced for DHSI 2014's *Fundamentals of Programming/Coding for Human(s|ists)* by John Simpson. Unlike most sheets that simply list commands followed by brief explanations this sheet aims to provide context first and possibly useful commands second.

REMEMBER

1. UNIX based commands are usually case sensitive.
2. Silence is Golden. If a command does not issue an error then it did what you told it to do (which may not be what you wanted it to do).
3. All directories are part of the same tree structure. This starts with a folder called "root", represented by a single "/".
4. Do not use spaces in filenames.
5. In the commands below everything in angle brackets (< and >) and the angle brackets themselves should be replaced with your terms.

Finding Things to Help Yourself

Find things to do	Type **help** to get a list of *some* basic built-in commands. Find out what each item listed does by typing **help** ⟨command⟩.
Find more commands	Want to search *all* the available commands for a word? Use **man -k** ⟨word⟩ for a list of all man(ual) pages containing <word>. Use **man** ⟨command⟩ to bring up an explanation for <command>. The arrow keys navigate the explanation. **h** gives instructions. **q** returns you to the command prompt.
Find files	Locate all files named <fname> by using **find / -name** ⟨fname⟩. Specify a higher directory than root (/) to speed up this command.
Find files contaning a word	Use **grep** ⟨word⟩ ⟨dir⟩ to find all files in directory <dir> that contain <word>.
Find where a program is located	**which** ⟨program⟩ will tell you where the program run by typing <program> is located. **whereis** ⟨program⟩ will tell you where the standard version of <program> is. Modifications to the system may lead these no not be the same.
Clean up	Use **clear** to wipe the workspace clean.

Chaining

Pass output of one command to another	The pipe character "|" (above the "\") can be used to make the results of one command the input of a second command. For example **history | grep rm | tail -12** will take the list of all the previous commands typed, pull out only those that included the text "rm" and then show only the last twelve of these. Try adding **sort -r**.
Save output of a command to a file	Use a single closing angle bracket (">") as you would the pipe character and make sure you end the chain with a filename. If the file already exists it will be overwritten and if not it will be created. Use double closing angle brackets (">>") to append output to an existing file (or create the file if it doesn't currently exist).

Navigating the tree-based file system

Find out where you are in the tree	Type **pwd** (print working directory) to see where you are in the directory structure.
Find out what is around you	Use the **ls** command to *list* all the files and directories in the working directory. modifiers add even more power. **ls -a** will show *all* the contents, even if hidden. **ls -l** will provide the long list of additional details. **ls -al** will do both.
Move around	You move around using one of four variants of the **cd** (change directory) followed by a directory location. There are four important modifiers: **cd ..** to move to the parent directory. **cd /**⟨someDir⟩**/**⟨anotherDir⟩**/**⟨etc.⟩ to move to a new directory from the root of tree. **cd** ⟨childDir⟩**/**⟨grandchildDir⟩**/**⟨etc.⟩ to move into a child directory of the current directory (note the lack of a slash at the beginning). **cd ~** to return to the home directory.

File Control

Peer at the contents of a file.	**cat** ⟨file⟩ gives the full contents of a file. **head -**⟨n⟩ ⟨file⟩ shows the first <n> lines of a file. **tail -**⟨n⟩ ⟨file⟩ shows the last <n>. **wc** ⟨file⟩ provides the number of lines, words, and bytes.
Edit a file	**nano** ⟨file⟩ . The window lists commands at the bottom. A "^" means the control key so hold down control and press "x" to exit. **tr 'a' 'b' <** ⟨file⟩ will replace each 'a' with a 'b' in <file>.
Copy a file	**cp** ⟨sourceFile⟩ ⟨targetFile⟩ will do this for you. You can include directory information with the file name to copy from and to directories other than the current one.
Move a file	**mv** ⟨sourceFile⟩ ⟨targetFile⟩ behaves like copy but without the duplication.
Create a link	**ln -s** ⟨originalFile⟩ ⟨link⟩ will create a soft pointer to <originalFile> from <link>.
Create a directory	**mkdir** ⟨dirLocation⟩ will build a new directory.
Delete a file or directory	**rm** ⟨filename⟩ will permanently delete <filename>. Sometimes you might have to get tough and use **rm -f** ⟨filename⟩ to *force* it. To delete a directory use **rm -d** ⟨dirname⟩. If you want to delete *everything* in a directory use **rm -fR** ⟨dirname⟩ to *recursively* force removal of all contents, including child directories.
Expand a zipfile	Most downloads will come as tarballs that have been (g)zipped (extension .tar.gz). Extract these using **tar -zxvf** ⟨filename⟩.

Poweruser Tricks

Act as the Superuser	Put **sudo** at the start of any command and give the root password. The superuser is able to do anything in the system so be careful!
Type faster	Use the tab key to auto-complete filenames.
Type less	Use the up/down arrows to recall commands. Can also try Control-R.
Slow scrolling	Pipe to **less** or **more** to stop rapid scrolling.
Surf the web	If installed, use **links** or **lynx** to browse with text!

Stop something from running	Control-C will cancel a process in the current window (like **find**). For others use **ps** or **top** to get the process id and then use **kill** ⟨pid⟩. If it still won't die then use **kill -9** ⟨pid⟩.
Download a file	**wget** ⟨url⟩. **wget -c** ⟨url⟩ will resume a stopped download.
Use wildcards	* will stand in for any combination of zero or more characters. ? stands for any single character and [xyz] stand for any of x, y, and z.

Just For Fun search the web for "funny Linux commands"