

# Housekeeping

- Course Resources  
**<http://bit.ly/1KSnAYm>**
- Washrooms  
**Outside and \_\_\_\_\_ ?**
- Pedagogical style  
**“The hard way”, hands-on, playful, and collaborative**
- Outside Resources  
**Software Carpentry and *Think Python***
- Sticky Notes  
**Help and feedback**

# Week At A Glance

	Morning	Afternoon
Monday	Welcome & Terminal Interface Introduciton	Practice Lab: Whales and Weasles
Tuesday	Programming Concepts Intro	Practice Lab: NLTK and other tools
Wednesday	Alt. Environments & Other Tricks	Project Selection & Planning
Thursday	Project Time	Project Time
Friday	Project Time & What's Next	Present Projects

Note that there will be some blurring...



# Day #1

## Getting Comfortable with the \*nix Command Line

John Simpson & Dennis Tenen  
Fundamentals of Programming/Coding for Human(s)ists

DHSI-2015

# What's Important Today

1. Quick history of \*nix
2. Learn basic terminal techniques
  1. Navigation
  2. Creation and Destruction
  3. Plumbing and Searching
3. Practice using those techniques

“Not only is UNIX dead,  
it’s starting to smell really bad.”

*–Rob Pike circa 1991  
(widely attributed but unsourced)*

Not only is it not dead, but Linux and OSX have further embedded it in computer culture. It may not be perfect, but it’s powerful and here to stay.





The Cathedral

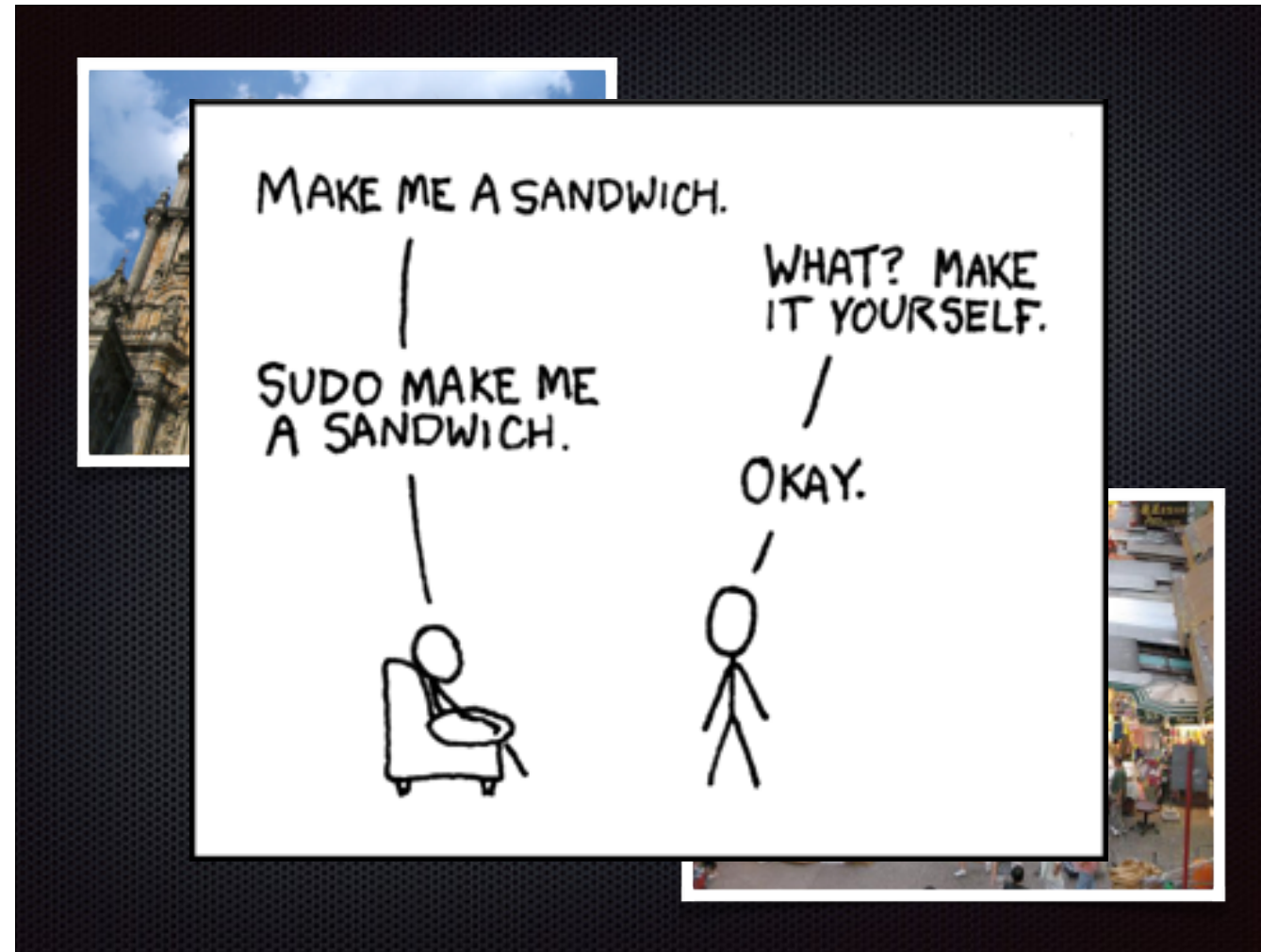


The Bazaar

Eric S. Raymond.

UNIX was written in C. C was written to write UNIX. (Not entirely true but a good way to think about it).

XKCD comic illustrates a core architectural difference between Windows and \*nix systems.



Eric S. Raymond.

UNIX was written in C. C was written to write UNIX. (Not entirely true but a good way to think about it).

XKCD comic illustrates a core architectural difference between Windows and \*nix systems.



Some of the major flavours of GNU/Linux available today.  
It is unfortunate for the history of the free software movement that “GNU” is so often dropped from the name (possibly because it is too hard to say--why is there no free marketing movement to prevent such things from happening?).  
Want to see *all* the flavours in distribution? Look [\*\*HERE\*\*](#).

Richard Stallman started the GNU project by quitting his job at the MIT lab when he could no longer have access to UNIX to create code for his printer.  
Built lots of stuff but didn't have a kernel for the operating system until Linus Torvalds built one.





Some of the major flavours of GNU/Linux available today.  
It is unfortunate for the history of the free software movement that “GNU” is so often dropped from the name (possibly because it is too hard to say--why is there no free marketing movement to prevent such things from happening?).  
Want to see *all* the flavours in distribution? Look **HERE**.

Richard Stallman started the GNU project by quitting his job at the MIT lab when he could no longer have access to UNIX to create code for his printer.  
Built lots of stuff but didn't have a kernel for the operating system until Linus Torvalds built one.

# Four things to keep in mind...

1. Silence is golden (or frustrating)
2. Capitalization matters
3. Spaces matter
4. Everything is a file

Spend a few minutes looking over this cheatsheet to get an idea of what can be done.

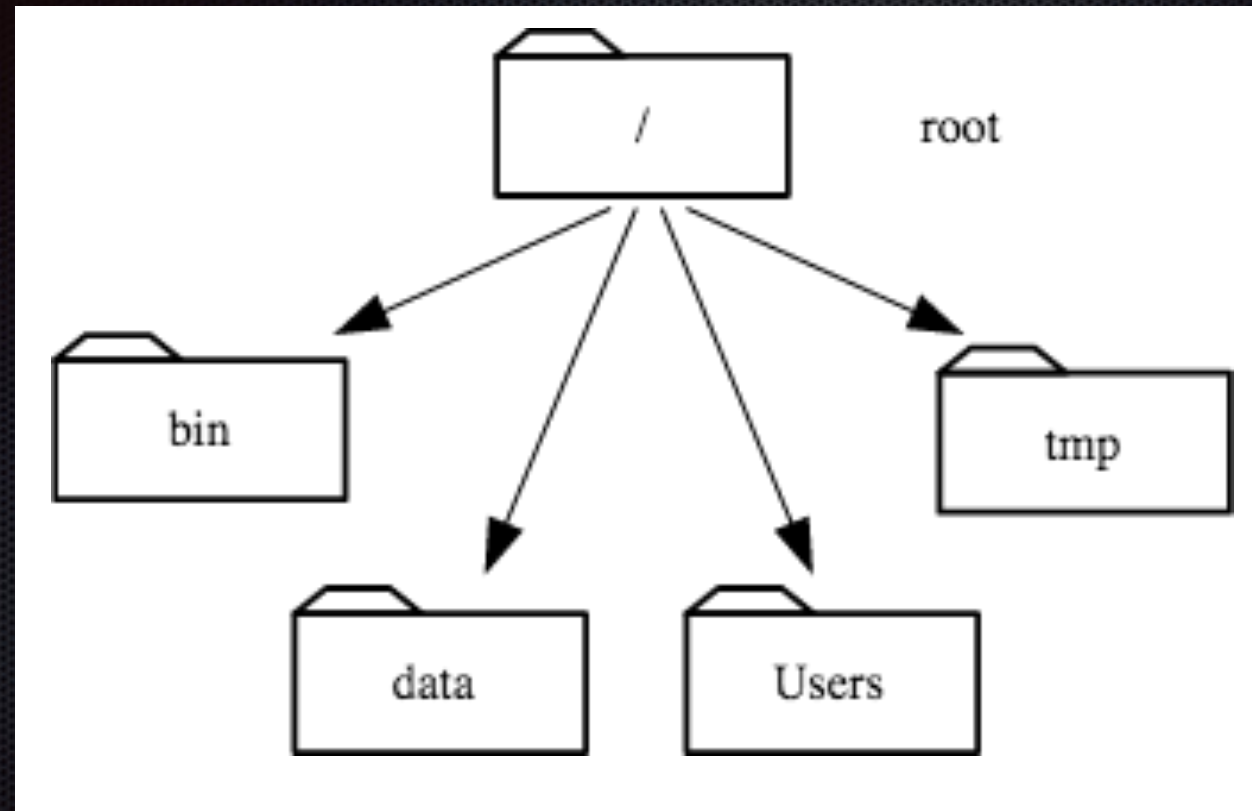
You can use it as a reference for what we'll do from here.

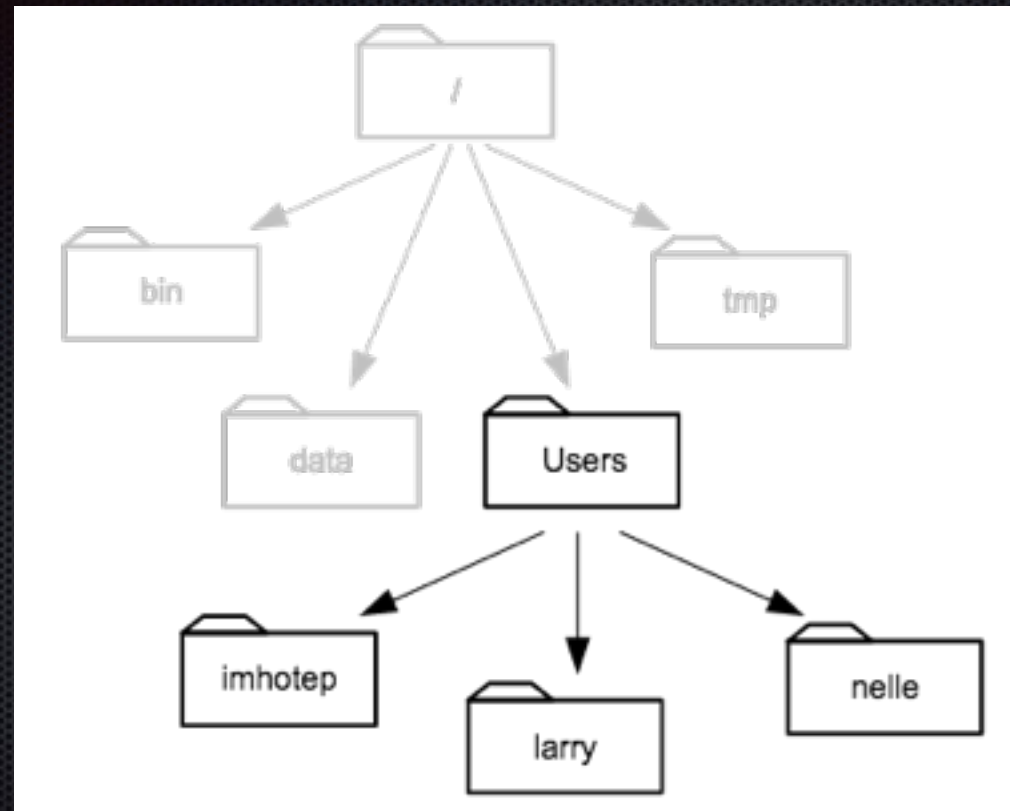
Basic UNIX Cheat Sheet	
<p><b>Introduction</b></p> <p>1. UNIX based commands are usually case sensitive.</p> <p>2. Spaces in filenames. If a command does not know an option then it did what you told it to do (which may not be what you wanted it to do).</p> <p>3. All directories are part of the same tree structure. This starts with a hidden called 'root' represented by a single '/'</p> <p>4. Do not use spaces in filenames.</p> <p>5. In the commands below everything in angle brackets (&lt; &gt;) and {} and the angle brackets themselves should be replaced with your terms.</p>	
<p><b>Finding Things to Help Yourself</b></p> <p>Find things to do: Type <b>help</b> to get a list of some basic built-in commands. Find out what each one does by typing <b>help command</b>.</p> <p>Find more commands: Want to search all the available commands for a particular term? Use <b>grep -i command</b> for a list of manually pages that contain 'command'. Use <b>grep -i command</b> to bring up an explanation for 'command'. Use the man page to complete the explanation. It will give you further instructions for the man page view or <b>man</b> will return you to the command prompt.</p> <p>Find files: Locate all files named 'filename' by using <b>find / -name filename</b>. Specify a higher directory than root (/) to speed up this command.</p> <p>Find files containing a word: Use <b>grep -i word file</b> to find all files in directory 'file' that contain 'word'.</p> <p>Find where a program is located: <b>which program</b> will tell you where the program can be found. <b>whereis program</b> will tell you where the standard version of 'program' is. Modifications to the system may lead these to not be the same.</p>	
<p><b>Changing</b></p> <p>Put output of one command to another: The pipe character ' ' allows the output of one command to be the input of a second command. For example <b>ls   grep .txt</b> will take the list of all the previous commands output and only those that end in '.txt' and then show only the first results of those.</p> <p>Save output of a command to a file: Use a single dotting angle bracket (&gt;) to save the output of a command and save it to a file and then show with cat. If the file already exists it will be overwritten and if not it will be created. Use double dotting angle brackets (&gt;&gt;) to append output to an existing file but create the file if it doesn't already exist.</p>	
<p><b>Power User Tricks</b></p> <p>Act as the superuser: The superuser is able to do pretty much anything in the system or only do this when necessary. That <b>sudo</b> at the start of any command. You will be asked for the root password before the command will execute.</p> <p>Copy files: Use the tab key to auto complete filenames.</p> <p>Copy files: Use the up/down arrows to recall commands. Can also use Ctrl-R.</p> <p>Run something: Type <b>Ctrl-C</b> to stop something from running.</p>	
<p><b>Navigation and File Control</b></p> <p>Find out where you are in the tree: Type <b>pwd</b> to get the working directory to see where you are in the directory structure.</p> <p>Find out what is around you: Use the <b>ls</b> command to list all the files and directories in the working directory. modifiers add even more power. <b>ls -l</b> will show all the contents, even if hidden. <b>ls -d</b> will provide the long list of additional details. <b>ls -a</b> will do both.</p> <p>Move around: Use <b>cd</b> to move around using one of three variants of the <b>cd</b> command. <b>cd ..</b> will move you to the parent folder of your current location (unless you are in /). <b>cd /</b> will move you to the root of the directory tree (from the shell at the beginning). <b>cd ~</b> will move you into a hidden directory of your current directory (note the lack of a slash at the beginning).</p> <p>File Control</p> <p>See file contents: Type <b>cat filename</b> to see the full contents of a file on screen. <b>head -n filename</b> will show the first n lines of file. <b>tail -n filename</b> will show the last n lines.</p> <p>Edit a file: <b>vi filename</b> is an easy and powerful option. The same modes help commands at the bottom of the screen. <b>q</b> means the control has to hold down control and press 'q' to exit.</p> <p>Copy a file: <b>cp filename destination</b> will do this for you. You can include directory information with the file name to copy from and to destination other than the current one.</p> <p>Move a file: <b>mv filename destination</b> behaves like copy but without the duplication.</p> <p>Create a link: <b>ln -s filename link</b> will create a soft pointer to 'filename' from 'link'.</p> <p>Create a directory: <b>mkdir filename</b> will build a new directory.</p> <p>Delete a file or directory: <b>rm filename</b> will permanently delete 'filename'. Sometimes you might have to get tough and use <b>rm -r filename</b> to force it. To delete a directory use <b>rm -rf filename</b>. If you want to delete everything in a directory use <b>rm -rf *</b> to delete everything in a directory.</p> <p>Export a file: Most downloads will come in a file that has been zipped (compression file). Extract these using <b>unzip filename</b>.</p> <p>Stop something from running: <b>Ctrl-C</b> will cancel a process in the current window (like <b>ls</b>). For other use <b>kill -9 pid</b> to get the process id and then use <b>kill pid</b>.</p> <p>Download a file: <b>wget filename</b> will download a file from the web.</p> <p>Use the web: If needed, use <b>curl</b> or <b>lynx</b> to browse with text.</p> <p>Use wildcards: '*' will stand in for any combination of one or more characters. '?' stands for any single character and [abc] will accept between a, b, and c.</p>	
<p><b>Just For Fun</b> search the web for 'UNIX' commands!</p>	



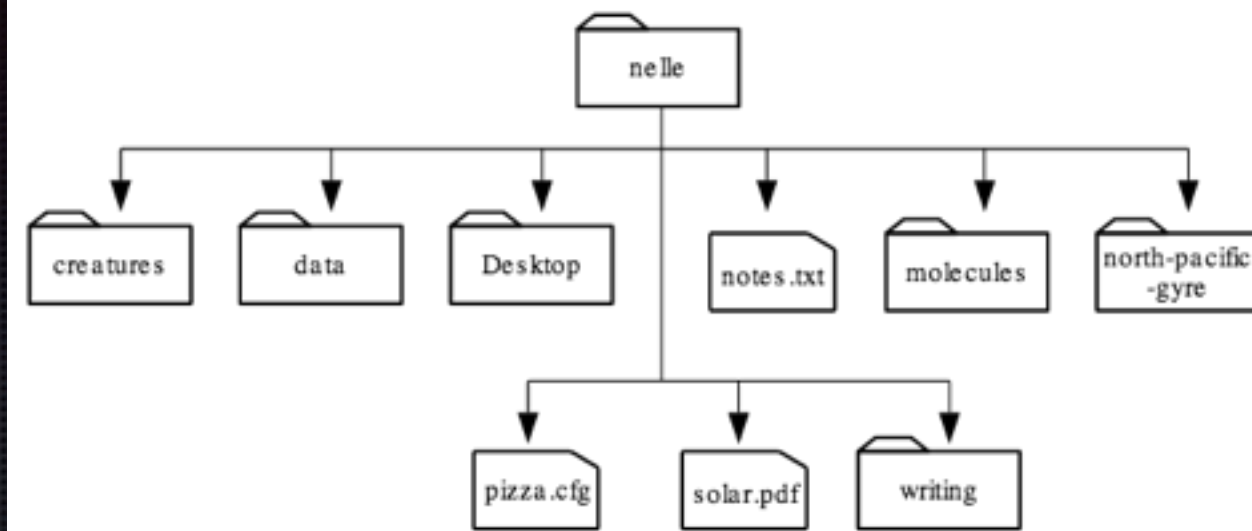
# Skills to Master: Navigation

- Translate an absolute path into a relative path and vice versa.
- Construct absolute and relative paths that identify specific files and directories and directory content
- Identify the actual command, flags, and filenames in a command-line call.
- Demonstrate the use of tab completion and up arrow review
- Demonstrate an ability to move around the file system









893366114733589115338836  
7421746485779902649474316  
4 2488156948038  
317923597279  
904  
293366114733589115338836  
0503  
3488115692880378  
20503  
17140924874967494  
8034608381  
923597270  
5823695701375058  
119492  
4671068124861169  
76003781140256820  
9825903624760242715

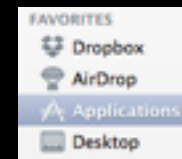
893366114733589115338836  
7421746485779902649474316  
4 2488156948038  
317923597279  
904  
293366114733589115338836  
0503  
3488115692880378  
20503  
17140924874967494  
8034608381  
923597270  
5823695701375058  
119492  
4671068124861169  
76003781140256820  
9825903624760242715



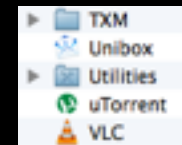
1. Open “Finder” from the list of icons at bottom of the screen (aka the “Dock”)



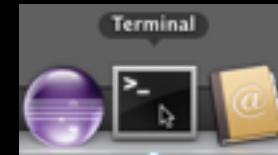
2. On the left of the new window is a category called “Favorites”. Choose “Applications”.



3. In the list of installed programs on the right find “Terminal” in the “Utilities” Folder



4. Drag “Terminal” to the Dock



5. Click on “Terminal”

# Skills to Master: Navigation

- Explain the similarities and differences between a file and a directory.
- Translate an absolute path into a relative path and vice versa.
- Construct absolute and relative paths that identify specific files and directories and directory content
- Explain the steps in the shell's read-run-print cycle.
- Identify the actual command, flags, and filenames in a command-line call.
- Demonstrate the use of tab completion, and explain its advantages.

# Skills to Master: Creation & Destruction

- Create a directory hierarchy that matches a given diagram.
- Create files in that hierarchy using an editor or by copying and renaming existing files.
- Display the contents of a directory using the command line.
- Delete specified files and/or directories.



# Skills to Master: Plumbing

- Redirect a command's output to a file.
- Process a file instead of keyboard input using redirection.
- Construct command pipelines with two or more stages.
- Explain what usually happens if a program or pipeline isn't given any input to process.
- Explain Unix's "small pieces, loosely joined" philosophy.

# Skills to Master: Searching

- Use `grep` to select lines from text files that match simple patterns.
- Use `find` to find files whose names match simple patterns.
- Use the output of one command as the command-line parameters to another command.
- Explain what is meant by “text” and “binary” files, and why many common tools don’t handle the latter well.



<http://bit.ly/1yIW533>