# 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

DHSI-2015

Fundamentals of Programming/Coding for Human(s|ists)

# 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…

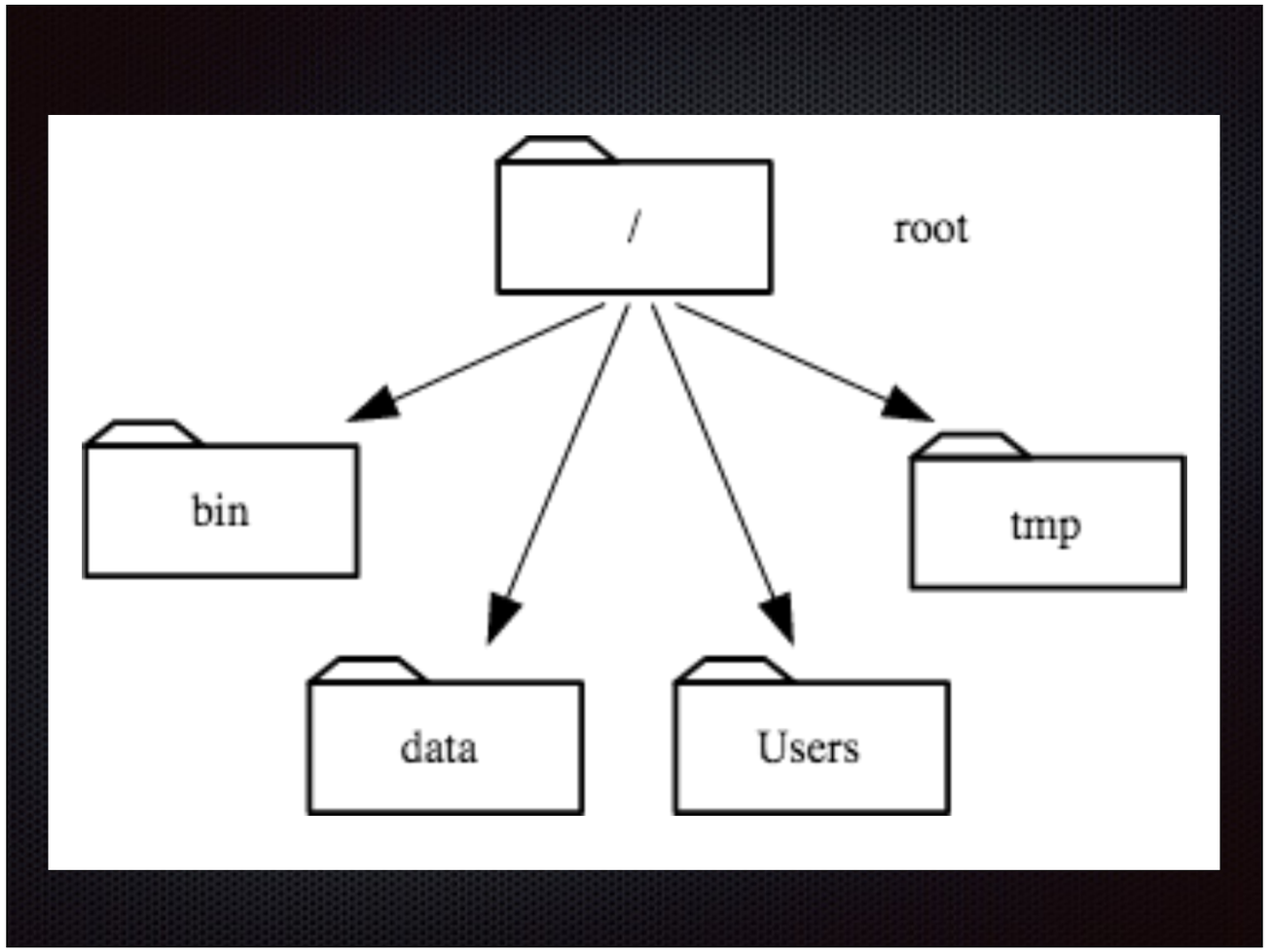
1. Silence is golden (or frustrating)
2. Capitalization matters
3. Spaces matter
4. Everything is a file

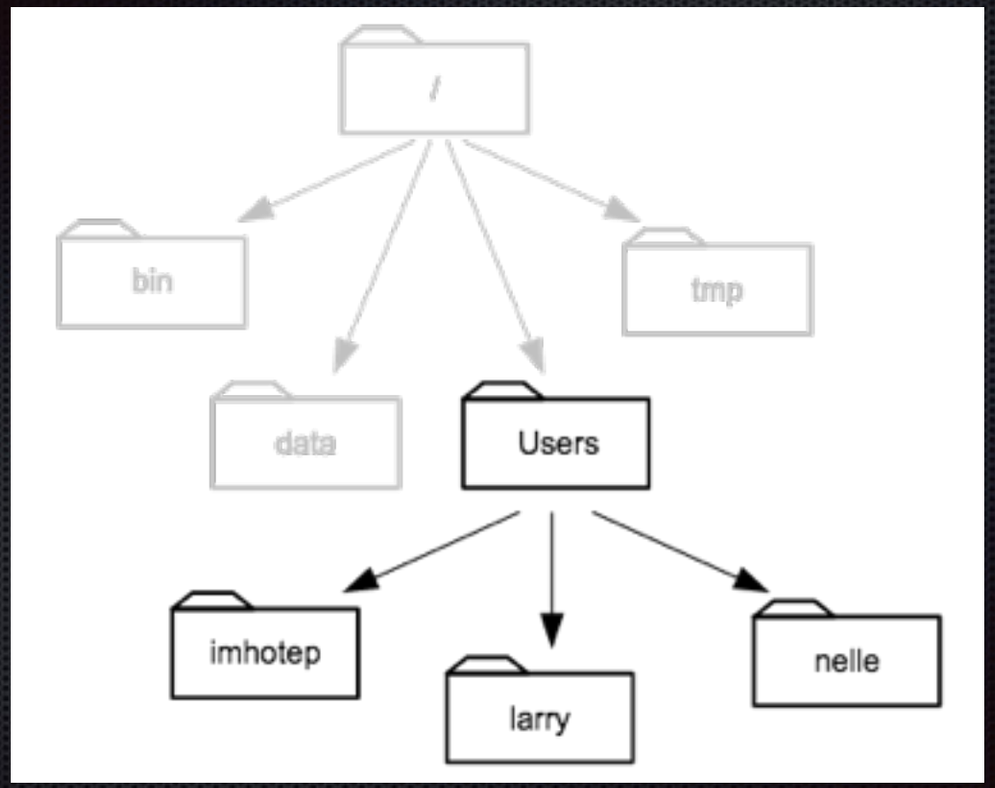
Spend a few minutes looking over this cheatsheet to get an idea of what can be done.

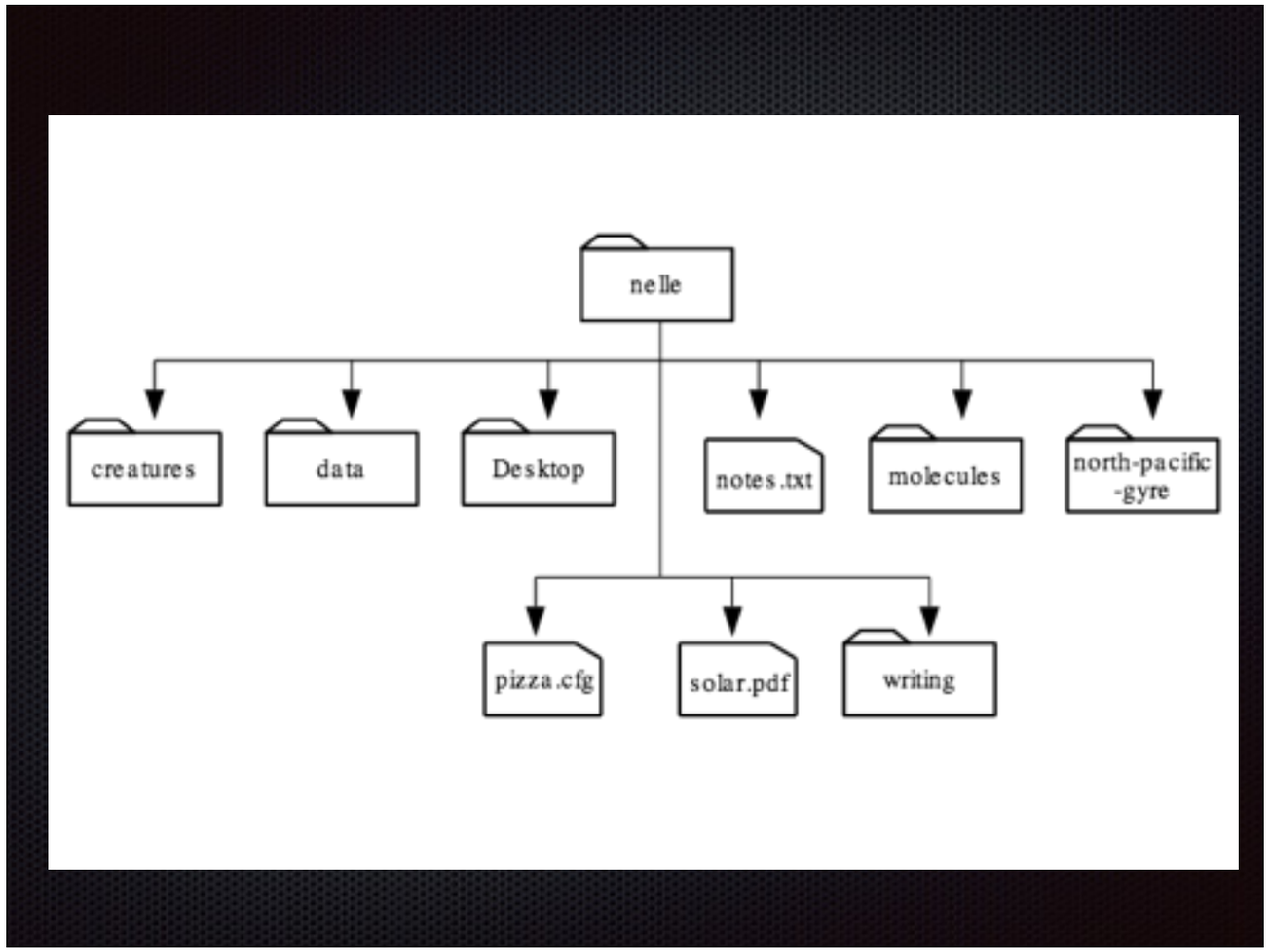
You can use it as a reference for what we'll do from here.

# 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

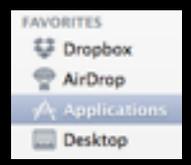

/
root
bin
tmp
data
Users


/
bin
tmp
data
Users
imhotep
larry
nelle

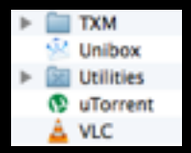

nelle
creatures
data
Desktop
notes.txt
molecules
north-pacific
-gyre
pizza.cfg
solar.pdf
writing

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

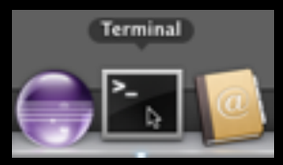
2. On the left of the new window is a category called “Favorites”. Choose “Applications”.




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 & Distruction

- 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/1ylW533