LINUX and the Art of Deep Cleaning

I. Interfaces

Most of us grew up on GUIs, short for Graphical User Interfaces- or rather our desktops full of icons representing programs and files. These relationships to our technology have evolved visually to meet our needs as humans over time, and interactions with our machines continue to evolve.

However, most modern computer users are unaware of how slow, and unspecified this process can be. It is because graphics are not the first language of the computer you're interacting with, here we will take our interactions with these machines back to their original human-computer interface, using something called a **Terminal.**

Well, beyond punch tape and switches. Let me introduce, the command line.

II. The Shell

The **shell** is a program that takes keyboard commands and passes them to the operating system to carry out. This is where we make our calls to the machine.

1. The command line will respond to anything you choose to input, if you don't know any commands..

    Experiment by typing random stuff!

2. But, Linux comes with many built in commands

    a. try: date
    b. try: cal

3. Moving around inside of the terminal is done with a few simple commands

    pwd : print working directory
    cd : call Directory
    ls : list contents of directory

4. The home directory contains your main user files, you can call it by typing:

cd ~

5. Relative pathnames allow you to move one level up in your directory hierarchy:

cd ..

6. When you need to know which directory you're in, the pwd command comes in handy

pwd
/Users/myComputer/

7. In order to find out what files are in your directory, use the l**s** command. You can also specify exactly which directory you ant to look in, for example our main Pictures folder

ls /Users/myComputer/Pictures
image.jpg
image.png
etc/

8. you can combine **ls** with files and arguments

-a : List all files, including those who's name begins with a period
-d : list the details of the directory, rather than its contents
-F : appends an indicator character such as a forward slash to indicate a directory (ex. bin/)
-h : display the file size in human readable format rather than bytes
-l : display results in long format
-r : display the results in reverse order
-S : Sort result by file size
-t : sort by modification times

For example, typing **ls -a** will display all files

7. If you want to see the contents of a text file, you can use the **less** command - (q - quit less)

less file.txt

III. Manipulating files and directories:

1. similar to the GUI, we have simple copy and move commands with with to organize our files

cp : copy file
mv : move file
mkdir : make directory
rm : remove file or directory

2. wildcards allow you to select

*** : all files**
? : any single character
[characters] : any character thats a member of set characters
[!characters] : any character that is not a member of set characters
[[:class:]] : any character that is a member of a specified class

3. examples of how to use wildcards

b* : any file beginning with b
c*.txt : any file beginning with c followed by any characters ending with .txt
cat?? : followed by exactly 2 characters
[abc]* : any file beginning with with a, b, or c

IV. The Simple Art of Organization

1. Naming conventions.

First we need to create a few directories to move all of our files to

mkdir _JPG

now lets look at the directory we just created **ls**

2. Next, let's copy all of the .jpg files over to the new JPGs folder using the ***** wildcard

cp *.jpg JPGs

3. To see where our files went, we can now

cd _JPG

4. Lets look at all of our images the way our machine interprets them

ls -a -l

5. Okay time to move more files, lets jump back to the parent directory

cd ..

6. Make two more folders, one for PNGs and another for PDFs

mkdir _PNG _PDF

7. Great! Now just like the JPG files, we can copy all of the .png and .pdf files over to their respective folders.

cp *.png _PNG
cp *.pdf _PDF

8. You can do this any file type, from here we can focus on the files that we need to organize, this is always done when good naming conventions are employed, but as the creator you get to decide what that means.

cd _JPG
mkdir cats
mv cats*.jpg cats

you can use the mv tool to rename files

- **mv (option) cat random\ garbage\ bad\ name.ext cat_ref_01.ext**

9. Finally, you can list all of the changes you have made, when you look at the directory, you will see a refreshingly clean set of folders holding those scattered files.