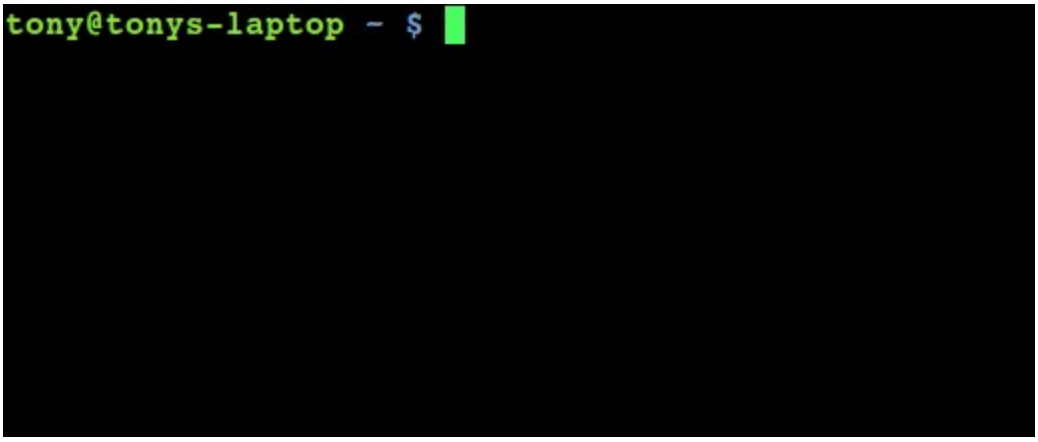
# Exploring the Shell

The Unix shell is very powerful and anything* you can do on your desktop by clicking through windows you can do by typing out a commands

**Getting your bearings**

You open a terminal and you are presented with this:




You start off in your users home folder.

Here are a few commands:

- ls → list contents of current folder
- find → recursively list all files from the current folder down
- cd → move to another folder
- mv → move files from one location to another
- cp → copy files from one location to another
  - dd → a very powerful copy command
    - `dd if=/dev/zero of=/dev/sda` will **completely** erase your harddrive if ran as root (though it will likely crash before finishing)

- touch → create a blank file
- mkdir → create a new folder
- nano → simple text editor

## Understanding the file system

The Unix filesystem is organized in certain way and understanding the organization will help greatly

- The highest point is the “/” or root folder
    - Everything that is your computer exists under this
    - Under “/” there are a few key folders
        - /bin → contains core system executables (echo, cd mv, etc...)
        - /etc → contaions configuration files
            - Instead of clicking through menus to configure a program you can edit certain files in this directory
        - /home → contains all user files
            - Each user has a folder under /home with all of their stuff
        - /var → contains log file as well as other constantly changing information
        - /proc → Is a virtual filesystem (not actually on the hard drive) that contains information about all running processes.
        - /dev → Are device files which represent the hardware of your computer
            - /dev/sda → is your harddrive
            - /dev/cdrom → is your cd rom drive

Now back to your home folder.

- Run `cd` to take you to your home folder
- Run `ls`
  - This prints the current folders contents
- Let's make a testing folder and move into it
  - `mkdir testing`
  - `cd testing`
- Run `ls` to show that folder is empty
- Run `touch test.txt` to create an empty file “test.txt”
- Run `ls` again
- Run `nano test.txt` to open an interactive text editor
  - Type some stuff and press Conrtol+X to exit
  - Press 'y' to save changes, then enter to confirm the name
- Run `cat test.txt` to view the contents of test.txt
- Run `echo “Hello!” >> test.txt`
  - “>>” is a redirection that appends to a file or creates a new one
  - “>” will overwrite a file or create a new one
- Run `cat test.txt` again to view it's new contents
- Run `cat test.txt > test.txt.backup`
  - This create a copy of test.txt called test.txt.backup, overwriting whatever was in test.txt.backup
  - It actually dumps the contents of test.txt to standard out, then redirects it to the file test.txt.backup
- Run `echo “Hello” > hello.txt`

- Then run `echo “World” > world.txt`
  - This will create a file hello.txt containing “Hello” and a file world.txt containing “World”
- Run `cat hello.txt world.txt > helloworld.txt`
  - This will append the contents of world.txt onto the contents of hello.txt in a new file called helloworld.txt
- Run `rm helloworld.txt` to delete this file
  - Be careful with rm as there is no undo
    - For shits and gigs run `rm –rf ~`
      - Just kidding, don't. That will delete everything under your home folder. Seriously don't do it.
- You can use wildcards to delete multiple files
  - Run rm *.txt to remove any files ending with .txt
- Run `touch newfile.txt`
- Run `mkdir newfolder`
- Run `mv newfile.txt newfolder/`
  - This will create a new file and folder then move the new file into the folder
- You can use the tab key auto fill commands
  - Type `ls newfo` then hit tab a couple of times and it should auto complete newfolder for you.

**This covered the basics of file management from the shell**

**Again there is a lot I haven't discussed, I recommend you:**

- **Check the man pages for every command you just learned.**
- **Experiment a little with the console, get comfortable there.**
- **Learn about how file permissions work in Unix**

**Understand that *everything* that is your computer is just a file somewhere in the filesystem**

- **/etc/crontab is a file that manages scheduled program executions**
- **/bin contains core utilities used by the system**
- **/dev/sda is the entire harddrive**
  - **Running `dd if=/dev/zero of=/dev/sda` will write zeros to your entire hard drive if ran as root (though it will crash before finishing if not ran from a live boot)**
    - **Don't try this at home**
- **/home/user contains all of *user*'s files**
- **/dev/null is where things go to die**