Using the Terminal
Open up the Terminal
Show what is in your current directory

- Think of the directory like a folder
- To show what’s in your directory/folder type in `ls` and then press enter
- This will show you all files and folders in the current folder
- This helps in knowing where to go next if you’re unsure of what’s in a folder
Visual representation of the `ls` command

Note: I am currently in a directory called “Root”, which is shown in Terminal after the colon
Changing directory

- To go to a subdirectory/folder, use the `cd` command
- Type in `cd [foldername]` to go into that folder
- `[foldername]` represents the name of the folder which you wish to go into

```bash
$ cd folder
```
Visual representation of the \texttt{cd} command

I am now in the directory folder, which was in the original Root folder.
Go up a directory

- To go back up the directory, you can also use `cd`
- Type in `cd ..`
- This will take you to the folder directly above current one in the hierarchy
- Typing that in in our example will take us back to the “Root” directory
Note on \texttt{cd} command

If you just type in \texttt{cd} and then press Enter/Return without a directory name, it will take you up to the top level of your directory hierarchy.

The end location varies depending on what type of machine you’re using, so the results won’t be the same for everybody.
Copying a file

- The **cp** command is used to copy files
- You type in `cp [oldfilename] [newfilename]` to make a copy of a file in the same directory
- `[oldfilename]` represents the name of the file that you want to make a copy of
- `[newfilename]` represents the name of the copied file (you cannot have two files of the same name in the same directory)
Visual representation of cp command

Note: file1.txt and newfile1.txt have the exact same content in them
Renaming files

- Renaming files uses the `mv` command
- Type in `mv [oldfilename] [newfilename]`
- Nomenclature is the same as before
Visual representation of renaming file

Note: newname.txt is the same as newfile1.txt, but with a different name
Moving a file

- Moving a file also uses the `mv` command.
- Rather than giving it a new name as before, you put in a location as your second value.
- Type `mv [filename] [location]` to move a file to a new location.
- In our example, we will move `newfile.txt` to the directory named `folder`.
- The `./` before the name is telling it that it is moving from the current directory to a subdirectory.
- You can also start from the root by using `~/directoryname`.

```bash
$ ls
file1.txt  file2.txt  folder
$ cd folder
$ cd ..
$ ls
file1.txt  newfile1.txt
$ mv newfile1.txt newname.txt
```

```bash
$ mv newname.txt ./folder
```
Visual representation of moving a file
Creating a new folder

- Creating new folders uses the **mkdir** command, which is short for *make directory*
- Type in `mkdir [directoryName]`
- Where `directoryName` will be the name you wish to give your new directory/folder
Visual representation of `mkdir` command
Seeing the new folder

If you type in the `ls` command, you will see the updated directory with all the current files.
Opening files for reading

- There are multiple ways to open a file.
- If you want to just read the file without editing it, you can use the `cat` command.
- Type in `cat [filename]` to do this.
Result

```
Ademola-MacBook-Pro:Root ademolaolyinka$ cat file1.txt
CS 216 is going to be awesome
Ademola-MacBook-Pro:Root ademolaolyinka$
```
More with **cat**

- You can also open up two files at once using `cat`
- To do this, type `cat [file1] [file2]`
- The example command and result are shown to the right
Editing files

- There are multiple ways to edit a file
- A very common and simple way to do this is using the **nano** command, which opens up a text editing program called GNU Nano
What Nano looks like

- I’ve added three exclamation points to my file
- The commands for use in nano are at the bottom and the carrot symbol (^) represents Ctrl (even on a Mac)
- e.g. to exit is Ctrl + X
- You cannot click here. You must use your arrow keys to navigate the file
- WriteOut is essentially the same as save. It will prompt you for a file name afterwards. If you want it to overwrite the old file, just keep the same name
- If I exit Nano and then use cat to read my file, it will have the three exclamation points added
Summary

These are the basics of using Terminal

After some practice, certain tasks will actually be easier using it (I didn’t believe it at first either)

More advanced use of it will come the more you need it

We will have an SSH tutorial later, which is the command used to access a remote server