

All Lessons / [Week 1 - Day 2](#)

# Practice What You Learned

## Terminal Practice In Breakout Rooms

### Episode X: A New Terminal

A long time ago in a Unix environment far, far away, young Jedi padawans who knew only of desktop software were seduced by the dark side of the Force to enter... The Terminal.

Follow the instructions below using all the console commands introduced in Fundamentals, class, or that you find on your own.

### Setup

- Open the **Terminal/HYPER app**
- Inside the `git-github-and-terminal` folder, create another folder called: `galaxy-far-far-away`
- Then create a file inside `galaxy-far-far-away` called `commands.txt`
- Paste the answer to each numbered question (i.e. the command(s) that accomplished the task) in `commands.txt` once you get it to work
- Remember, you can learn about any Unix command by typing `man` and then the command name. E.g., `man ls`. Type `Q` to get out of the Manual page ("man page") viewer

### Part I: Set the Scene

Complete all work inside the `galaxy-far-far-away` folder.

1. Create a directory called `death_star`, and make the following files inside of it: `darth_vader.txt`, `princess_leia.txt`, `storm_trooper.txt`
2. In `galaxy-far-far-away`, make a directory named `tatooine` and create the following files in it: `luke.txt`, `ben_kenobi.txt`
3. Inside of `tatooine` make a directory called `millenium_falcon`, and in it create: `han_solo.txt`, `chewbacca.txt`

### Part II: `mv` - rename

You can rename a file using the `mv` command.

4. Rename `ben_kenobi.txt` to `obi_wan.txt`

## Part II: `cp` - copy

You can copy a file from one location to another using the `cp` command. ( `man cp` for more info)

- Directories can be sibling (parallel to each other) or can be parents (the folder that contains the folder you are in)
- Copy `storm_trooper.txt` from `death_star` to `tatooine`

## Part IV: `mv` - move

You can use the `mv` command to move files from one location to another. `mv` can be used for renaming, moving, or both. Run `man mv` to see the options—remember hit the `Q` key to get out of the manual page viewer.

6. Move `luke.txt` and `obi_wan.txt` to the `millenium_falcon`
7. Move `millenium_falcon` out of `tatooine` and into `galaxy-far-far-away`
8. Move `millenium_falcon` into `death_star`
9. Move `princess_leia.txt` into the `millenium_falcon`

## Part V: `rm` - remove

**BE CAREFUL WITH `rm` !!! THERE IS NO "TRASH" IN THE UNIX CLI. WHEN YOU DELETE SOMETHING IT IS GONE FOREVER!!!**

You can use `rm` to delete a file.

10. Delete `obi_wan.txt` .

## Part VI: all together

11. In `galaxy-far-far-away` , make a directory called `yavin_4`
12. Move the `millenium_falcon` out of the `death_star` and into `yavin_4`
13. Make a directory in `yavin_4` called `x_wing`
14. Move `princess_leia.txt` to `yavin_4` and `luke.txt` to `x_wing`
15. Move the `millenium_falcon` and `x_wing` out of `yavin_4` and into `galaxy-far-far-away`
16. In `death_star` , create directories for `tie_fighter_1` , `tie_fighter_2` and `tie_fighter_3`
17. Move `darth_vader.txt` into `tie_fighter_1`
18. Make a copy of `storm_trooper.txt` in both `tie_fighter_2` and `tie_fighter_3`
19. Move all of the `tie_fighters` out of the `death_star` and into `galaxy-far-far-away`

## Part VII: `rm -r` : remove directories and everything they contain

**BE CAREFUL WITH `rm -r` THERE IS NO TRASH CAN IN THE UNIX CLI. WHEN YOU DELETE SOMETHING IT IS GONE FOREVER**

Before you hit enter, make sure are deleting the right thing, or you could accidentally delete the contents of your computer (it has happened).

This command will not typically ask you if you "really want to delete." It will just delete.

20. Remove `tie_fighter_2` and `tie_fighter_3`

## Part VIII:

21. Touch a file in `x_wing` called `the_force.txt`

22. Destroy the `death_star` and anyone inside of it

23. Return `x_wing` and the `millenium_falcon` to `yavin_4`

**Celebrate. You've reached the end of this homework :)**

## Commit and push your updated code:

"Add" your changes (prepare them to be "committed"):

```
1 | $ git add -A
```

"Commit" your changes—any time you make a commit, you can always restore the files in the repo to that point:

```
1 | $ git commit -m "Completed homework"
```

"Push" your commits to github:

```
1 | $ git push origin master
```

## Conclusion

You will "git" plenty of practice as we progress through this program, so if the concept of git/github still seems a little fuzzy at this point, rest assured you will soon "git" it once you "git" some more practice in. 😎