Managing Files

- `tree [dir]` command will give a visual representation of a directory structure
- The directory structure for this repo at the time this note was being written:

```
tree .

.
├── access-control-lists
├── files
│   ├── brace-expansion.md
│   ├── extended-globs.md
│   ├── file-globs.md
│   ├── files.md
│   ├── getting-information-on-files.md
│   └── pdf
│       ├── brace-expansion.pdf
│       ├── extended-globs.pdf
│       ├── file-globs.pdf
│       ├── files.pdf
│       └── getting-information-on-files.pdf
├── filesystem
│   ├── filesystem-layout.md
│   ├── README.md
│   └── standard-unix-filesystem-hierarchy.png
├── managing-files
│   └── managing-files.md
├── permissions
└── README.md
```

- to determine where you are in the filesystem use `pwd` - print working directory
- change directory with `cd [dir]`
- `cd ~` is a shortcut to `/home/user`
- `..` is the parent directory and you can move up a directory with `cd ..`
- `cd -` will always take you back to the last working directory
- Absolute paths always describe a location with reference to root `/`
- Relative paths always describe a location with reference to the current working directory `.`
    - When typing a relative path it is not strictly necessary to reference the current working directory. You can simply leave off the `.` for most things

Creating files and directories

- To create a directory use `mkdir [path]`
    - use the `-p` option for a parent directory when you're going to create nested dirs
        - ie. `mkdir one/two/three`
- You can create files from the terminal with vim, nano or gedit
- You can also create empty files with `touch [file]`

    - You can `echo` contents into a file with `echo "text" > [file]`
        - This will create the file and put the contents in it if the file doesn't already exist
        - If it does exist, it will overwrite the file so beware
        - To append contents to the end of the file use `>>` instead of `>`

Coping files and directories

- to copy a file or dir use `cp [options] [target] [destination]`
    - specify a new name in the destination path if you want to copy and change the name
    - use the `-r` option to copy directories recursively
    - use `-a` option to preserve metadata when copying

Moving and renaming files and directories

- to move a file or dir use `mv [options] [file] [destination]`
    - to change the name of the file in moving, specify a new name
    - if you do not wish to change the name, specify a destination path only
- to change the name of a file use the `mv` command with the working dir as the destination and a new name

Deleting files and directories

- to delete an empty directory use `rmdir [dir]`
- to delete a directory and all it's contents use `rm -r [dir]`
    - to force delete all contents add the `-f` option
- there is no trash can on the terminal. when you delete a file or directory, it's gone forever and requires special software to recover

Creating links

- Links provide shortcuts to organize your workspace
- Symbolic links and hard links
- To make a hard link run `ln [target file] [link]`
    - Hard links are like creating a second door to the same room
    - has the same inode number, points to the same data on disk
    - takes no space
    - doesn't break when target is deleted
    - A file that has a hard link will have two of the exact same inode number
    - disadvantages:
        - can only hardlink files
        - cannot hardlink across file systems
        - transparent - difficult to identify
        - must remove all inodes to delete hard links
- To create a symbolic link run `ln -s [target] [link]`
    - symlinks show up with `ls` with a point to the original file
    - advantages
        - link across filesystems
        - link to directories

        - easy to identify
    - disadvantages
        - take up a small amount of space
        - break if target is deleted
        - not seamless - commands may act differently with them
- Important note: when you delete a symlink, it cannot have a trailing `/` or you will delete the target directory, not the link

Making file manipulation safe

- by default, system will usually not warn you when you're going to do something destructive
- the `-i` option can be passed to `cp`, `mv` and `rm` to make the process interactive and ask for your confirmation before doing something
- you can make interactive mode default by editing the `.bashrc` file by adding the following to the bottom of the file:

```
alias cp='cp -i'
alias mv='mv -i'
alias rm='rm -i'
```

-Or, you can just pass the option when you need it