



The Terminal

FEAR NOT THE COMMAND LINE

// FLATIRON SCHOOL



LEARNING OBJECTIVES

- // Utilize bash commands through a terminal interface
- // Use the terminal to list, make, move and remove files and directories
- // Use the terminal to navigate between files/directories and open Jupyter notebooks or other files
- // Edit text files using vim

Terminal? Shell? Command Line?

- Many terms - all different but similar
 - Ultimately: we use the **Command Line** to enter text prompts and interact with the **Shell** interface, which is run by the **Terminal**
 - Realistically: these terms are often used interchangeably
- In the Flatiron Data Science program, we use:
 - Terminal Programs:
 - Mac - **Terminal** application
 - Windows - **Git Bash**
 - Shell options: **bash** / **zsh**

Basic Commands

\$ pwd

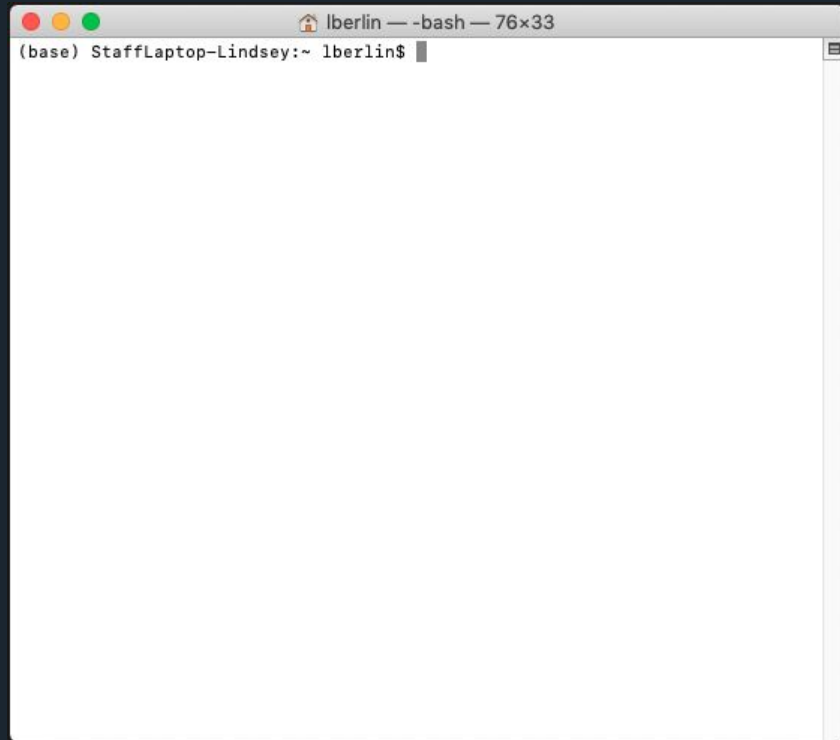
display the current working directory of the shell

\$ ls

list the files and directories of the current directory

\$ cd

change the directory to update the current working directory



Basic Commands

\$ touch

create a new file

\$ mkdir

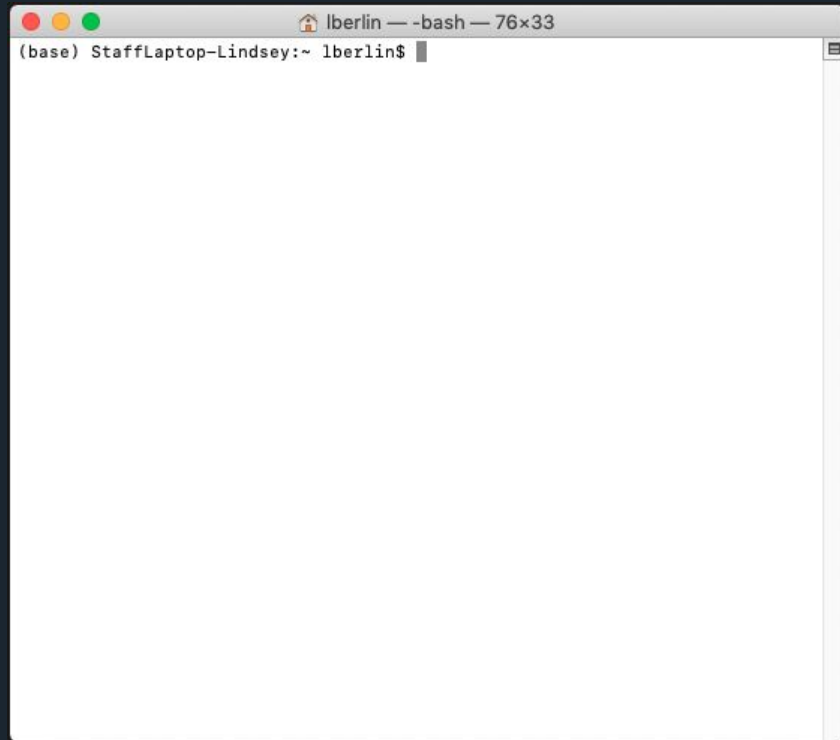
create a new directory

\$ mv

move a file from source to destination
(also used to rename files/directories)

\$ rm

remove a file from the file system
(BE CAREFUL!)



Special Directories

/

root, the top-level directory

~

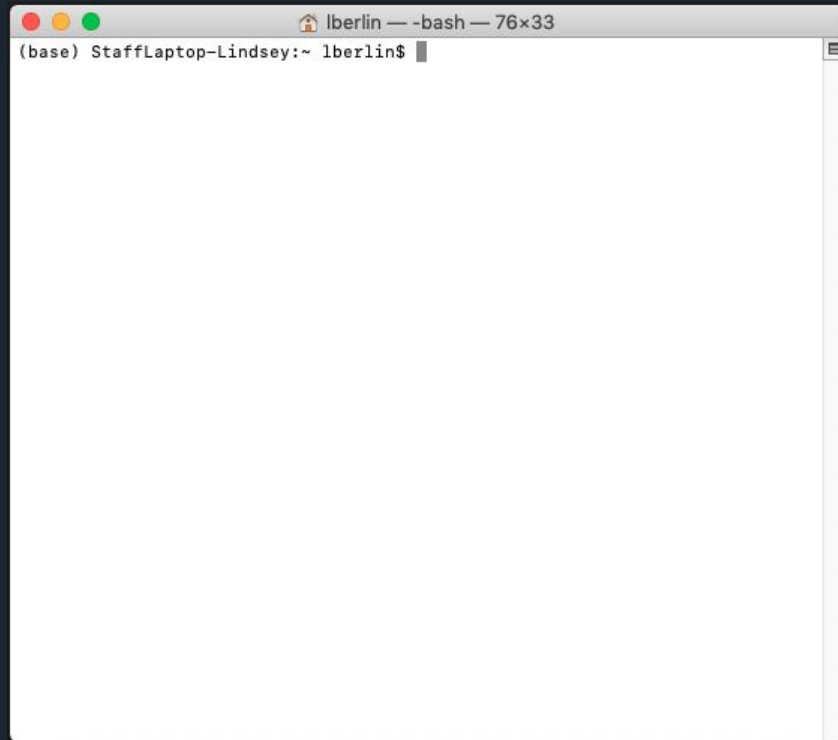
your home directory

.

the current directory

..

the parent directory (one level up)



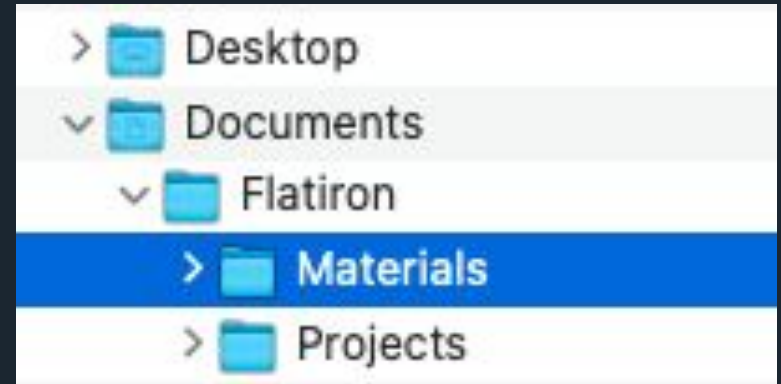
Paths - Absolute or Relative?

- **Absolute:**

starts from root (/) or home (~)

- **Relative:**

starts from your current working directory (where you are)



Prompt:

Given the file directory structure pictured above, what are the two versions of the path to the **Projects** folder, if you're currently in **Materials**?

Prompt: Make Your Flatiron Folder!

- Using only the Terminal, make a Flatiron folder where you can keep all program-related files and materials (if you haven't already)
 - Suggestion: Put it somewhere logical! In Documents or Desktop, perhaps
- Practice opening a new Terminal window and navigating to that folder!



LET'S GET
ORGANIZED

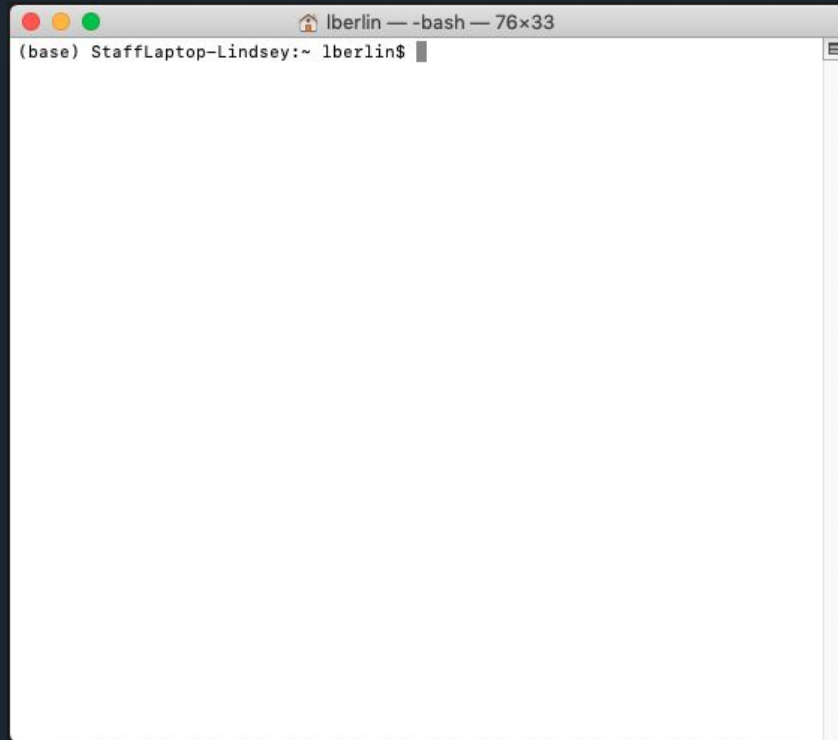
Text Editors

- Nice to use a GUI (graphical user interface) code-focused text editor
 - No matter which you use, configure that text editor so it can open easily from the command line!
 - If you use VS Code:
 - `code .` : open the current working directory
 - `code <FILENAME>` : open that file
 - (Macs: need to set up)
- Sometimes, you have to use a CLI text editor... enter **VIM**

Surviving VIM

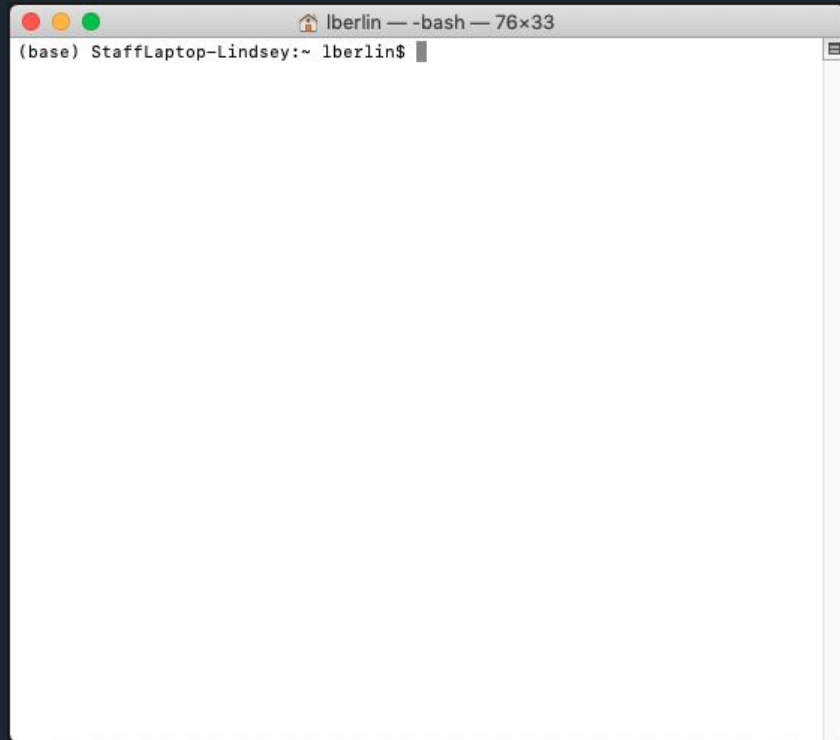
Two Modes:

- **Insert** mode
 - Type normally to add/edit text
 - Enter by pressing `i`
- **Command** mode
 - Each key is a command
 - Enter by pressing `ESC` key



Basic VIM Commands (used in Command mode)

i	enter Insert mode
A	enter Insert mode at the end of the line
ESC	return to Command mode
dd	delete the current line
u	undo last change
:wq	save and quit
:q!	force quit without saving



Additional Resources

Initial Learning Resources:

- OpenClassrooms' [course on the command line](#)
- MIT's [Terminus](#) command line game
- [Linux Commands Cheat Sheet](#)

Going Further:

- Unix Primer tutorial: [Basic Commands in the Unix Shell](#)
- Data Camp tutorial: [8 Useful Shell Commands for Data Science](#)