



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 (print working directory)
display the current working directory of
the shell

\$ ls (list)
list the files and directories of the current
directory

\$ cd (change directory)
change the directory to update the
current working directory

Special Directories

- / root, the top-level directory
 - DO NOT mess around here
- ~ your home directory
 - typically the 'user' level
- . the current directory
- .. the parent directory (one level up)

Paths - Absolute or Relative?

- **Absolute:**

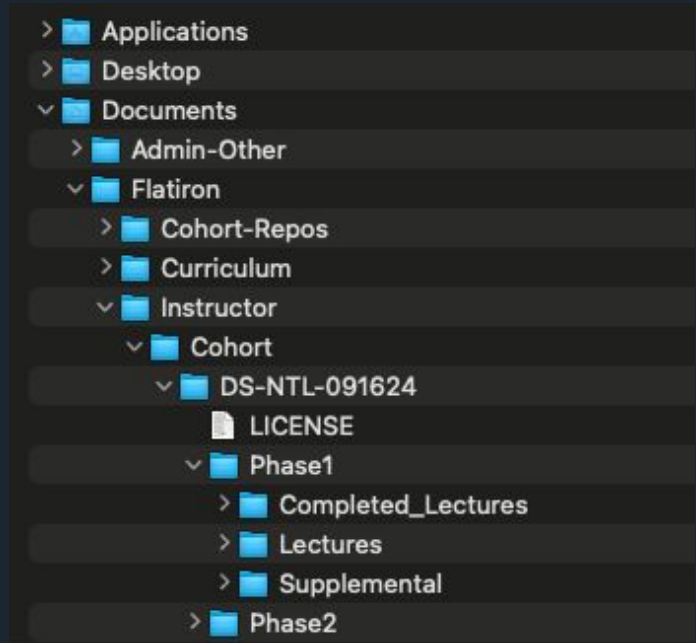
starts from root (/) or home (~)

- **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 **Phase2** folder, if you're currently in **Phase1/Lectures/**?



Basic Commands

\$ touch

create a new file based on extension

\$ mkdir

create a new directory/folder

\$ mv

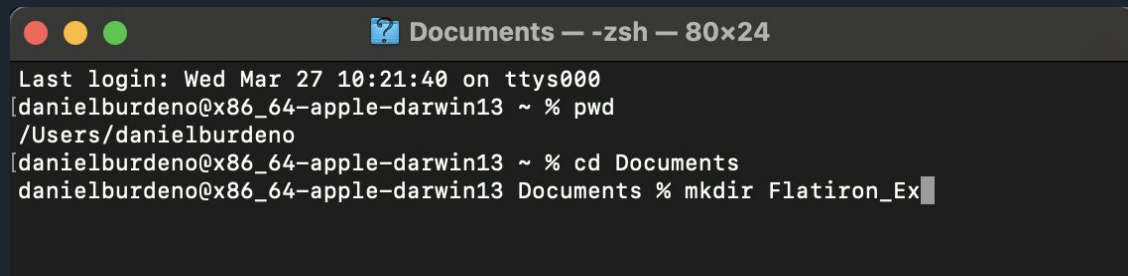
move a file from source to destination
(also used to rename files/directories)

\$ rm

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

Prompt: Make Your Flatiron Folder!

- Using 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



```
Documents — -zsh — 80x24
Last login: Wed Mar 27 10:21:40 on ttys000
[danielburdeno@x86_64-apple-darwin13 ~ % pwd
/Users/danielburdeno
[danielburdeno@x86_64-apple-darwin13 ~ % cd Documents
[danielburdeno@x86_64-apple-darwin13 Documents % mkdir Flatiron_Exchange
```

LET'S GET
ORGANIZED

Prompt: Make Your Flatiron Folder!

- Let's practice creating and moving files as well
 - Create a file named "text_example.txt"

```
[danielburdeno@x86_64-apple-darwin13 ~ % cd Documents
[danielburdeno@x86_64-apple-darwin13 Documents % mkdir Flatiron_Ex
[danielburdeno@x86_64-apple-darwin13 Documents % cd Flatiron_Ex
[danielburdeno@x86_64-apple-darwin13 Flatiron_Ex % ls
[danielburdeno@x86_64-apple-darwin13 Flatiron_Ex % touch text_example.txt
[danielburdeno@x86_64-apple-darwin13 Flatiron_Ex % ls
text_example.txt
[danielburdeno@x86_64-apple-darwin13 Flatiron_Ex % █
```

- Move that file up one folder

```
[danielburdeno@x86_64-apple-darwin13 Flatiron_Ex % mv text_example.txt ..
[danielburdeno@x86_64-apple-darwin13 Flatiron_Ex % ls
[danielburdeno@x86_64-apple-darwin13 Flatiron_Ex % cd ..
[danielburdeno@x86_64-apple-darwin13 Documents % ls
Admin-Other      Flatiron_Ex      Zoom
Flatiron         My Tableau Repository  text_example.txt
[danielburdeno@x86_64-apple-darwin13 Documents % █
```

- Go ahead and move it back!

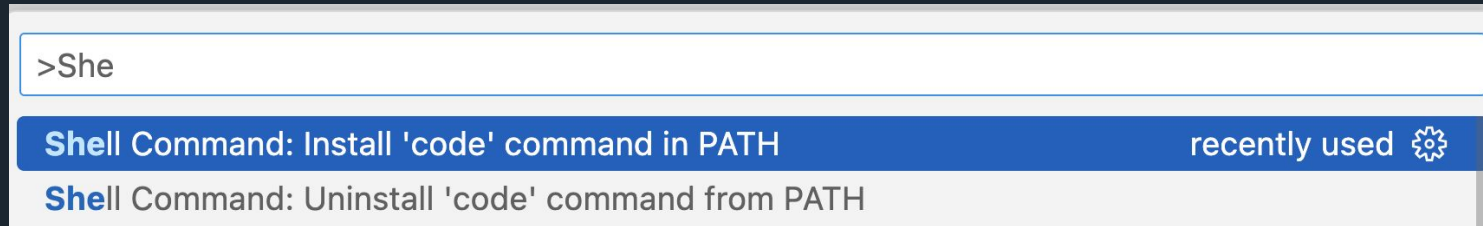
LET'S GET
ORGANIZED

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!
 - We will download VS Code - Windows users should already have
 - 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**

VS Code - Code Shortcut

- Microsoft Product - IDE
 - Able to create files/notebooks
 - Able to interact with GitHub
- Up Arrow + Command + P (Mac)
- CTRL + Shift + P (Windows)
 - Type ">Shell Command"
 - Click Install 'code' command in PATH
- Troubleshoot: Click uninstall first, quit VS Code, then install



Surviving VIM

Two Modes:

- **Insert** mode
 - Type normally to add/edit text
 - Access by pressing `i`
- **Command** mode
 - Each key is a command
 - Allows to save and exit
 - 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](#)
- Tips and Trick from a Flatiron Alum:
<https://www.realdifferencedata.com/2022-03-16-terminal-tips-and-tricks/>