Please download git, if you don't already have it:

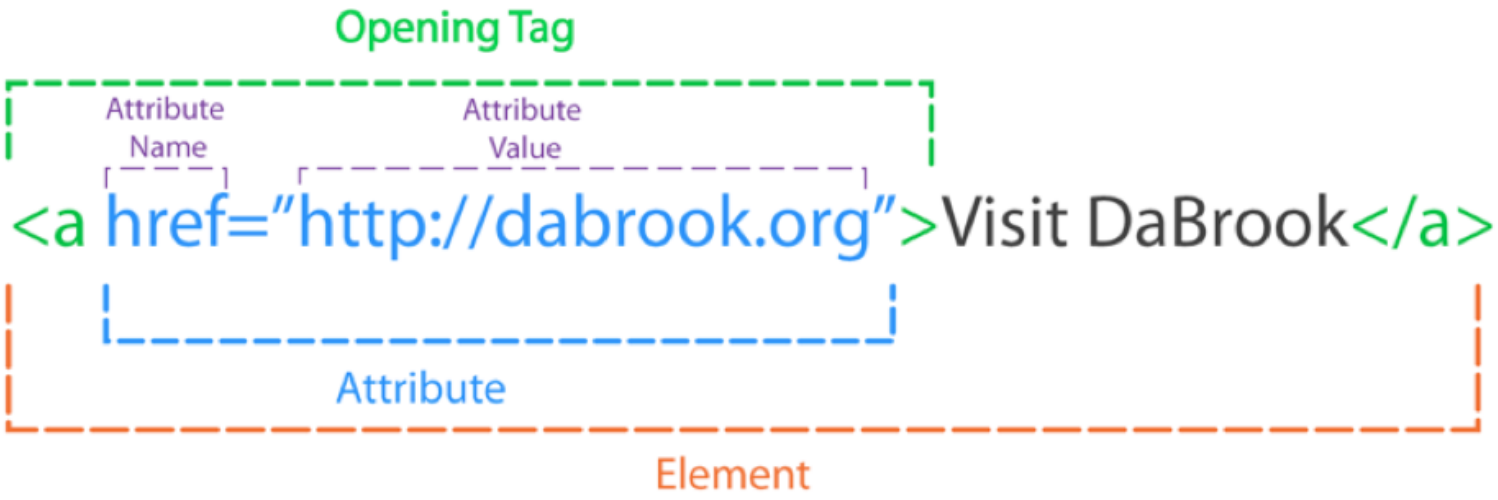
http://git-scm.com/downloads

If you are a Windows user, use Git Bash for all Terminal interactions. It is included with Git. To launch, Start Menu -> Programs -> Git -> Git Bash

WD100

The Terminal, Git

HTML Review


Opening Tag
Attribute Name
Attribute Value
Visit DaBrook
Attribute
Element

The Terminal - In Popular Culture!

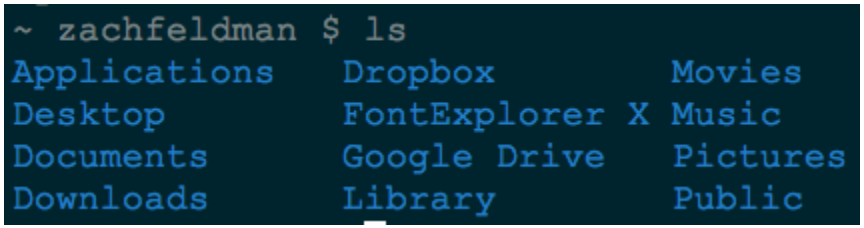
https://www.youtube.com/watch?v=Z4MGwdKPds8

The Terminal
(aka The Command Line)

- A faster way of interacting with your computer
- Lets you navigate directories, modify files, run programs, and much more!
- A few different types – you're probably using bash

`ls` - list directory contents

- Prints out a list of files in your directory



`cd` – change directory

- Used to switch to another directory

- Usage:

```
$ cd ~/Downloads/nycda-iwdd-lesson-2
$ cd ..  #Go up one directory
$ cd -   #Go to the previous directory
```

`pwd` – print working directory

- Print out the name of the directory you're currently in

```
nycda-new zachfeldman (master)$ cd ~
~ zachfeldman $ pwd
/Users/zachfeldman
```

`rm` - remove

- Delete a file

- Be careful with this one! It deletes *forever* – files don’t go to your Trash

Command Line Flags

- Many terminal commands have options - flags are used to pass these options to the commands

- For instance, with the rm command, you can append -r to delete recursively and to delete directories. You could also append -f to force deletion.

```
$ rm -rf <path> #be careful with this!
```

Git and Github

- Git – version control
- Github – a site to put your git repos

What is git?

- Think of Git like Mac OS X's "Time Machine"...for your code!
- A Git repository, or repo, is a project folder with a hidden .git folder to save history
- Each time you make a git "commit", you make a snapshot of your code that can be rolled back to at any time
- You'll need to learn git to deploy apps to anywhere besides your local machine

Basic Git Flow

```
$ git init
# for new projects to initialize a git repository, only
 done once per folder/project


$ git add -A
# add all recent changes that have been made to be
 “staged” for a commit


$ git commit -m “Initial commit”
# make your first git commit for the project


$ git status
# make sure the working directory is “clean” meaning all
 changes have been committed, or just see at what stage
 in the process you’re at
```

Git Tips

- Make sure you only call `$ git init` once! Don't forget that it's only to initialize a new git repository in a project folder

- If you're unsure about whether a project has a git repository or not, you can just call `$ git status` before you do anything, no harm done

- `$ git add -A` stages all outstanding changes, `$ git add <filename>` just stages that file. If you're ever unsure what's being staged for this commit, call `$ git status`

Basic Git Commands

git status	See if files have been modified, new files have been created, and what files have been staged for a commit
git log	show a log of all recent commits, use q to quit the log
git add <file or folder name>	add only a specified file or folder to your git repo
git add -A	add all changes and files to a commit
git commit -m “Some commit message”	make a “commit”, a snapshot of the current state of your project

Git Tips II

- Generally, make commits whenever a feature or part is done, or just whenever you're at a point you'd like to be able to roll back to in the future

- Use descriptive commit messages, not just "changes" but instead "`Added the 'about us' link in the header`"

- Practice using git by creating a folder with one file in it, then trying to navigate to it, initialize a git repository, make a commit, make a change, make a commit etc. until it's second nature. Keep practicing!!