git-fundamentals

A starting point for discovering the wonderful world of Git, GitHub, and Git Annex (Assistant)

Getting the tools / Setup

- Sign up for github (https://github.com), and **verify your email**.
- Command line tools (you'll need this!):
    - GitHub recommends the official Git installer (http://git-scm.com/downloads)
    - **I** recommend homebrew (http://brew.sh) for hackers on OS X
    - And it's easier to use your package manager (apt, yum, ...) for Unix, etc.
- GUI tools
    - Windows & Mac, also supports hg: SourceTree (http://www.sourcetreeapp.com/)
    - Lightweight Mac client: GitX-dev (http://rowanj.github.io/gitx/)
    - Lightweight GTK client: gitg (https://wiki.gnome.org/Apps/Gitg) (get it with your package manager).
    - Integrated with file manager:
        - TortoiseGit (https://code.google.com/p/tortoisegit/) on Windows
        - RabbitVCS (http://rabbitvcs.org/) on Linux (also integrates with gedit)
    - Many others (http://git-scm.com/downloads/guis) are available, and you can search for even more!
- Integration for your text editor!
    - Vim fugitive (https://github.com/tpope/vim-fugitive)
    - Emacs (http://www.emacswiki.org/emacs/Git)
    - Sublime Text 2 (https://github.com/kemayo/sublime-text-git)
    - gedit (use RabbitVCS)

Awesome resources

- Basic commands cheat sheet(s) (http://git-scm.com/docs) (Note the links to other cheat sheets!)
- Gamified git (http://pcottle.github.io/learnGitBranching/)
- GitHub sponsors this (http://try.github.io/).
- Where does "help" go from GitHub? Here (https://help.github.com/).
- @davclark is an awesome resource, himself! (when he's not being a jerk)
- Linux commands cheat sheet (http://www.pixelbeat.org/cmdline.html) (also mostly works for OS X command line).
- git - the simple guide (http://rogerdudler.github.io/git-guide/)
- The visual git guide (http://marklodato.github.io/visual-git-guide/index-en.html)

Local Git expositors at UC Berkeley

- A problem-focused exploration of git features is available in @jkitzes' Data

Science Lessons (http://jkitzes.github.io/datasci-lessons/)
- An IPython notebook (https://github.com/fperez/reprosw/blob/master/Version%20Control.ipynb) (of course!) from @fperez.
- Lots of exposition (http://matthew-brett.github.io/pydagogue) on git + other mostly python stuff and python scripts for a standardized git workflow (https://github.com/matthew-brett/gitwash) from @matthew-brett.
- Tons of R code to access public APIs (http://ropensci.org/packages/index.html) from @karthik and the @rOpenSci team.

Fancy GitHub features

Map diffs are amazing (you may have to click the "rich diff" button as opposed to the raw "<>" button):

- Explore the history of markers for restaurants (https://github.com/DU-GIS/Geojson_Data/blob/master/Restaurants.geojson)
- See how congressional redistricting looks (https://github.com/benbalter/congressional-districts/commit/2233c76ca5bb059582d796f053775d8859198ec5?short_path=85d2c1b#diff-85d2c1b78193e963475250414e57940b)

CSV diffs are strangely not as good, but (properly formatted) CSV is displayed nicely and is searchable:

- A tables of S&P 500 securities (https://github.com/datasets/s-and-p-500-companies/blob/master/data/constituents-financials.csv)

Git Annex

Read more here: http://git-annex.branchable.com/

Forking! What's that?

Forking on github is same as cloning in the command line.

Branching is also very important for collaboration

For branching, you should check the above link for now.