Introduction to GitHub

Julia Stewart Lowndes, PhD Project Scientist, Ocean Health Index National Center for Ecological Analysis and Synthesis, UC Santa Barbara lowndes@nceas.ucsb.edu

What is GitHub?

GitHub is an open-source development platform that enables easy collaboration and versioning, which means that all saved versions are archived and attributed to each user. It is possible to revert back to any previous version, which is incredibly useful to not only to document what work has been done, but how it differs from work done in the past, and who is responsible for the changes. Great for collaborators and your future self.

What is GitHub?

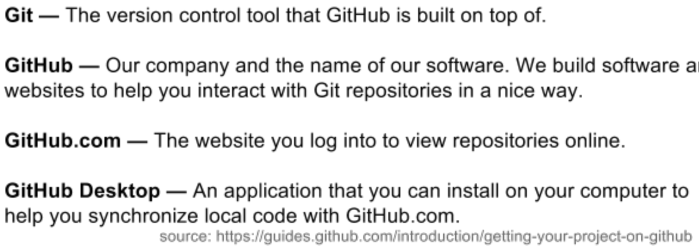
GitHub is an open-source development platform that enables easy collaboration and versioning, which means that all saved versions are archived and attributed to each user. It is possible to revert back to any previous version, which is incredibly useful to not only to document what work has been done, but how it differs from work done in the past, and who is responsible for the changes. Great for collaborators and your future self.

Similar to Dropbox, you have certain folders on your local computer that will be 'watched', with any changes able to be synced online. However, with GitHub, you have more control about what is synced, and how often. You can store, share, track changes and collaboratively edit many filetypes (including this presentation!) using any application. There are many other great features, including a to-do list you can share with collaborators (called Issues).

git and GitHub

version control system that tracks changes to your work

online platform to organize/store your work + collaborate


Git — The version control tool that GitHub is built on top of.

GitHub — Our company and the name of our software. We build software and websites to help you interact with Git repositories in a nice way.

GitHub.com — The website you log into to view repositories online.

GitHub Desktop — An application that you can install on your computer to help you synchronize local code with GitHub.com.

source: https://guides.github.com/introduction/getting-your-project-on-github

Git is distinct from GitHub, but this is not the focus today.

(definition from GitHub.com)

Why use GitHub?

There are so many reasons to use GitHub. Personal organization, switching between your home and work computers, backing up, version control. Also for collaboration, sharing, learning, contributing. . .

Nicely explained by Hadley Wickham: Git and GitHub Tutorial

And also by Karl Broman: GitHub Tutorial

And also by Ben Best: ds-git Tutorial

Workshop Outline

1. GitHub Structure
2. GitHub Workflow and Vocabulary
3. Best Practices
4. Syncing Options
5. Workflow and Practice
6. Resources

GitHub Structure

Files are stored in **repositories**, owned by **users** / **organizations**.

Repositories (‘repos’) are essentially folders containing files pertaining to a specific project. Repositories are version controlled so that any modifications to files, additions or deletions, are tracked and attributed to contributors with the correct permissions.

GitHub Structure

Files are stored in **repositories**, owned by **users** / **organizations**.

Repositories (‘repos’) are essentially folders containing files pertaining to a specific project. Repositories are version controlled so that any modifications to files, additions or deletions, are tracked and attributed to contributors with the correct permissions.

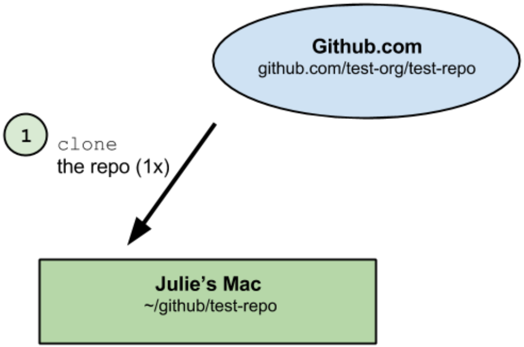
Same structure across all orgs/repos: familiar, easy to navigate.

Let’s explore a bit:

- **repositories**: github-intro, dplyr, ggplot2
- **users**: jules32, hadley, jennybc
- **organizations**: twitter, netflix, rstudio, nceas, ohi-science

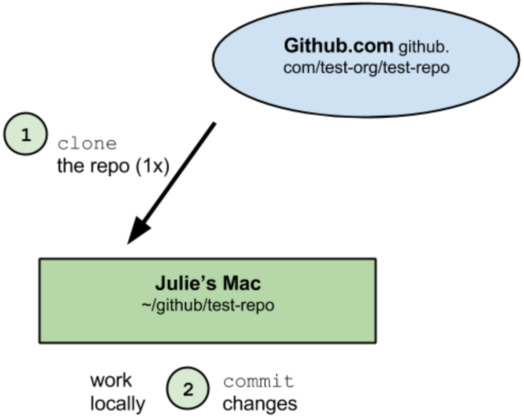
GitHub Workflow and Vocabulary

- **clone**: download to your computer from online version with syncing capabilities enabled



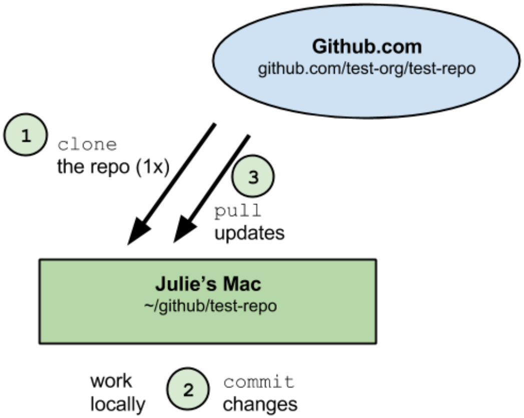
GitHub Workflow and Vocabulary

- **commit**: message associated with your changes (best practices)


Github.com github.com/test-org/test-repo
1 clone the repo (1x)
Julie's Mac ~/github/test-repo
work locally
2 commit changes

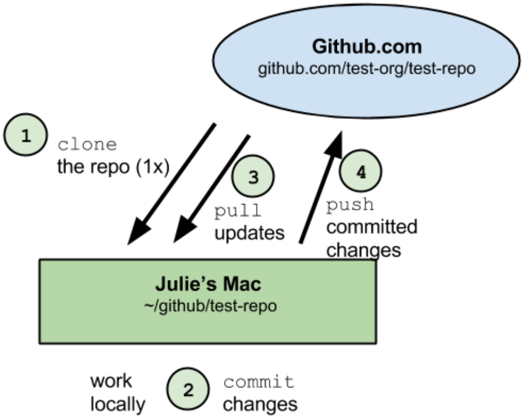
GitHub Workflow and Vocabulary

- **pull**: sync a repo on your computer with the online version. Do this frequently.



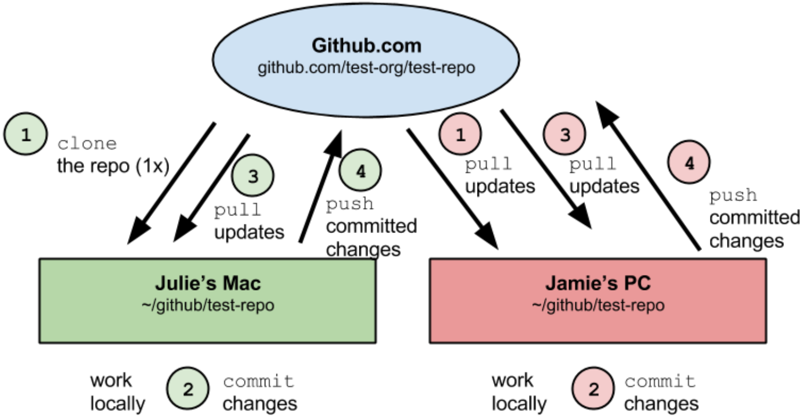
GitHub Workflow and Vocabulary

- **push**: sync the online repo with your version, only possible after committing



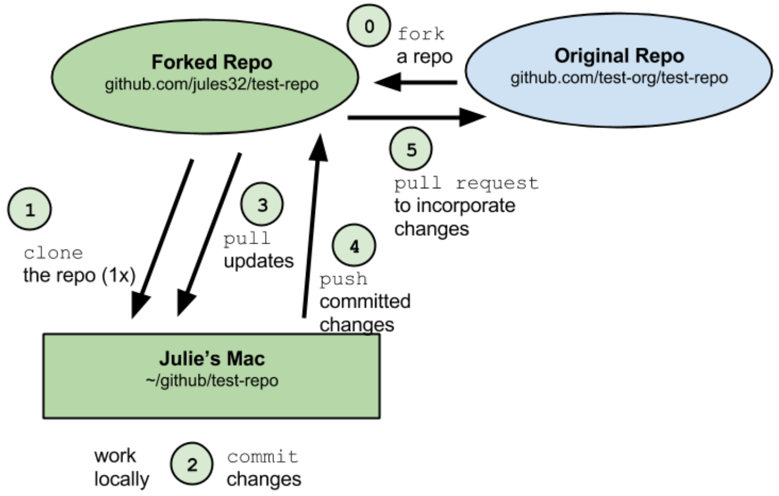
GitHub Workflow and Vocabulary

sync ∼ pull + commit + pull + push
All collaborators work independently but sync regularly

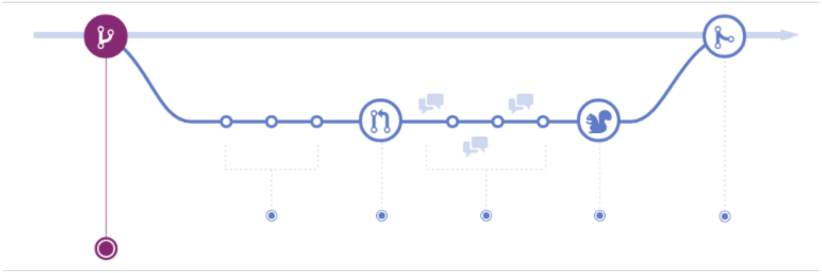


GitHub Workflow: fork and pull

fork + pull + commit + push + pull request



GitHub Workflow: branches and issues



guides.github.com/introduction/flow

Best Practices

Pull often!

Commit frequently

Be mindful of filepaths

- We work from a in a folder in our home directory called ***'github'*** (all lowercase!), so that everyone can access the repo with the filepath beginning in `~/github`:
- **Windows**: `Users\[User]\Documents\github\`
- **Mac**: `Users/[User]/github/`

Syncing Options

On GitHub.com, you can clone a repo to your computer.

On your computer, you can clone/sync repos in several ways:

- **GitHub Desktop**
- **RStudio**
- **shell/command line**

When you work on your computer, any edits you make to any files in your repo, using any program, will be tracked.

Creating new repos

On GitHub.com, you can create new repos

On your computer, you can create new repos in several ways:

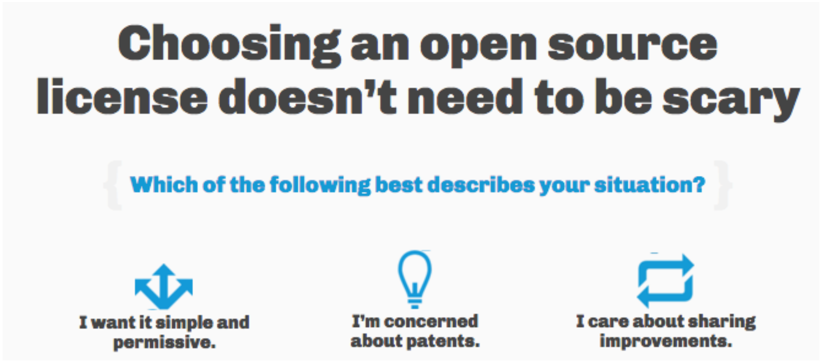
- **GitHub Desktop**
- **RStudio**
- **shell/command line**

Workflow and Practice Using RStudio

1. **fork a repo** to your user account
2. **clone the repo** to your computer
3. **edit a file**, inspect differences
4. **commit** changes
5. **pull**
6. **push**
7. repeat!

GitHub and science (and beyond)

- GitHub gets its science on
- Making your code citable
    - Choosing an open source license



Choosing an open source license doesn't need to be scary

Which of the following best describes your situation?

I want it simple and permissive.

I'm concerned about patents.

I care about sharing improvements.

- Ways to use GitHub that aren't coding

Resources

Learn more about GitHub:

- **GitHub Guides** by GitHub
- **Git and GitHub** by Hadley Wickham
- **Good Resources for Learning Git and GitHub** by GitHub
- **Learn Git Branching** by Peter Cottle
- **Git/GitHub Guide** by Karl Broman
- **Git & GitHub** by Ben Best

Just Google 'GitHub Tutorial. . . '