

## Introduction to GitHub

Alexa Fredston-Hermann PhD Student, Bren School  
fredstonhermann@ucsb.edu Special thanks to Dr. Julie  
Stewart Lowndes who created this entire presentation

# Does this look familiar






























	thesis draft 4.29 v2	Apr 30, 2012, 8:03 AM
	thesis for realsies.2	Apr 29, 2012, 11:54 PM
	thesis for realsies.3	Apr 29, 2012, 11:54 PM
	thesis for realsies	Apr 29, 2012, 11:54 PM
	thesis for real this time	Apr 29, 2012, 11:50 PM
	thesis for real this time .2	Apr 29, 2012, 11:50 PM
	thesis for real this time .3	Apr 29, 2012, 11:50 PM
	fredston-hermann thesis to combine	Apr 29, 2012, 11:44 PM
	fredston-hermann thesis to combine.2	Apr 29, 2012, 11:44 PM
	fredston-hermann thesis to combine.3	Apr 29, 2012, 11:44 PM
	fredston-hermann thesis.2	Apr 29, 2012, 11:33 PM
	fredston-hermann senior thesis.4	Apr 29, 2012, 11:32 PM
	fredston-hermann senior thesis.3	Apr 29, 2012, 11:31 PM
	fredston-hermann senior thesis.2	Apr 29, 2012, 11:29 PM
	bibliography	Apr 29, 2012, 10:52 PM
	thesis draft 4.29	Apr 29, 2012, 5:53 PM
	thesis draft 4.278	Apr 29, 2012, 7:48 AM
	thesis draft 4.27	Apr 27, 2012, 1:40 PM
	thesis draft 4.25	Apr 26, 2012, 11:00 PM
	thesis draft 4.23	Apr 25, 2012, 9:50 AM
	thesis draft 4.19	Apr 23, 2012, 6:14 PM
	thesis draft 4.16 aaron comments.docx	Apr 20, 2012, 2:07 PM
	genera research 4.9.12.docx	Apr 19, 2012, 11:35 AM
	thesis draft 4.16	Apr 18, 2012, 1:31 AM
	thesis draft 4.15	Apr 15, 2012, 7:43 AM
	thesis draft 4.13	Apr 14, 2012, 11:07 PM
	thesis draft 4.5.12	Apr 4, 2012, 11:42 PM
	thesis draft 4.2.12	Apr 2, 2012, 9:12 PM
	introduction 4.2.12	Apr 2, 2012, 8:36 PM

Figure 1:

# 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>

Figure 2:

# 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.

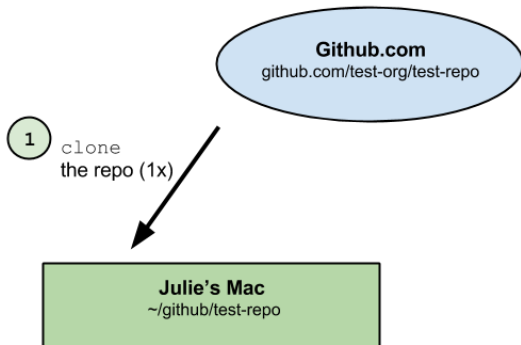
Same structure across all orgs/repos: familiar, easy to navigate.

*Let's explore a bit:*

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

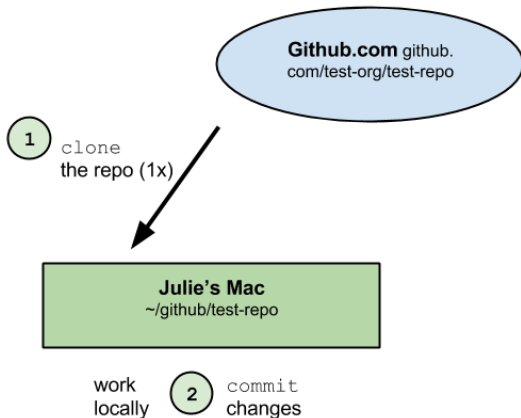
# GitHub Workflow and Vocabulary

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



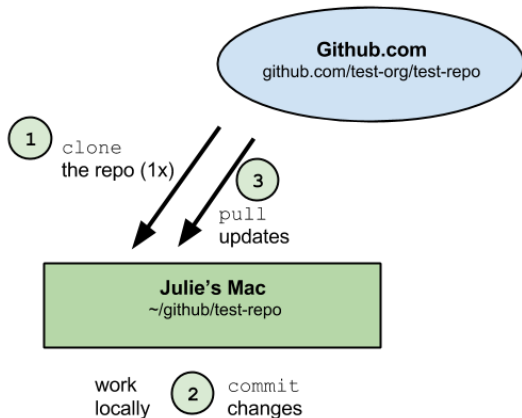
# GitHub Workflow and Vocabulary

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



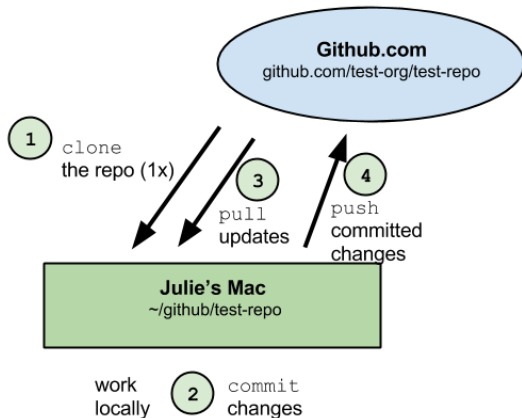
# GitHub Workflow and Vocabulary

- **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

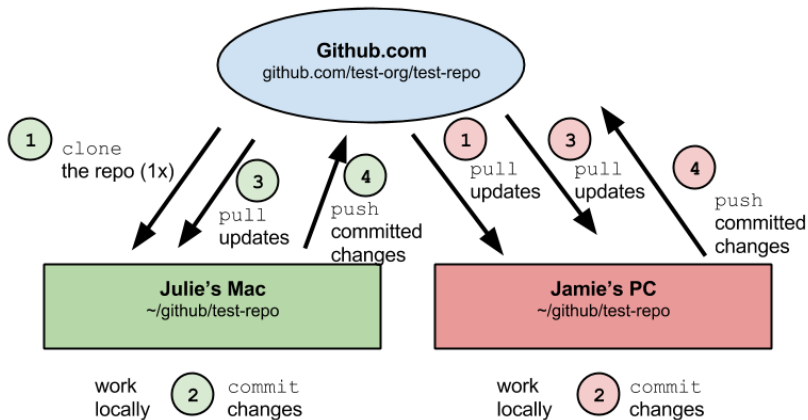


Figure 7:

# GitHub Workflow: fork and pull

fork + pull + commit + push + pull request

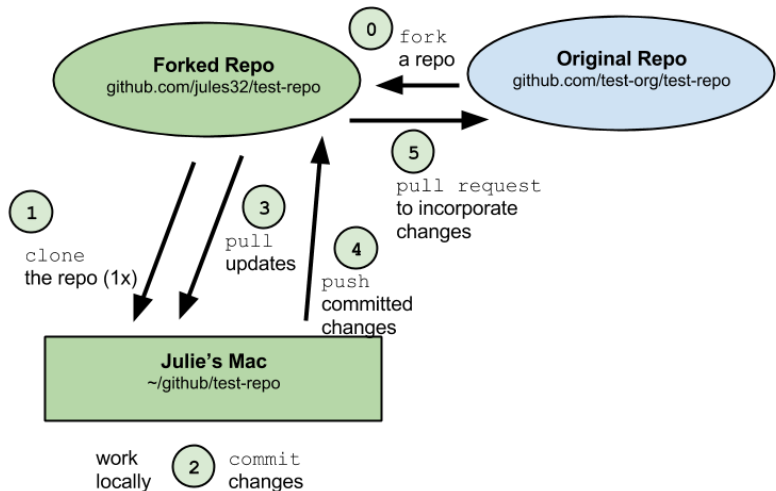


Figure 8:

# GitHub Workflow: branches and issues

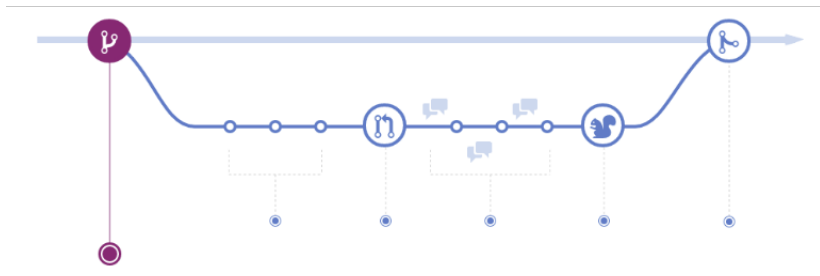


Figure 9:

[guides.github.com/introduction/flow](https://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.**

Figure 10:

# 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...'