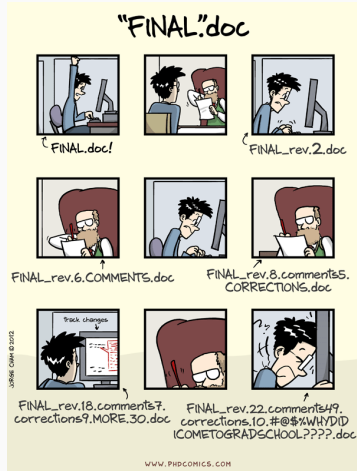


Git: Version Control

Arvind K. Saibaba

North Carolina State University

Motivation



These slides owe a great deal to <http://swcarpentry.github.io/git-novice/>.

Reasons to use Github

- Version control.
 - Safely go back to a working version of the code/document. Track changes.
 - Not just for code; use it for books, thesis, etc.
- Group projects.
 - Allows users to simultaneously work on the same project, and a mechanism to merge conflicts.
- Sharing code with the public.
 - Release code/ software packages
- Allows you to host a website.
 - Use Github pages + Markdown. Easy to use templates.
 - See this page for an example.

1. Google Drive/Dropbox
2. Google Documents (Pages, Sheets, etc.)
3. Microsoft Word/Open Office (Track changes)
4. Overleaf
5. SVN
6. Mercurial

Getting started

Setting up an account

- Public versions
 - GitHub, GitLab, BitBucket.
 - GitHub only allows free public accounts.
- NC State has a Github Enterprise Account.
 - Unlimited number of private repositories.
 - Public accounts are open only to NC state users.

Ways to use git

- Command line.
- Website.
- Github desktop account.
- Linked to other services such as overleaf.

Getting started

Setting up an account

- Public versions
 - GitHub, GitLab, BitBucket.
 - GitHub only allows free public accounts.
- NC State has a Github Enterprise Account.
 - Unlimited number of private repositories.
 - Public accounts are open only to NC state users.

Ways to use git

- Command line.
- Website.
- Github desktop account.
- Linked to other services such as overleaf.

Principles of git

Repositories

- There is a central repository (hosted by github/bitbucket, etc.)
- There are local copies (possibly several)

Work flow:

- Make your changes in your local copy
- Decide which changes you want to keep
- Synchronize with the central repository - Resolve conflicts if necessary

Each time this synchronization occurs, git stores a “snapshot” of the versions. This way you have a complete history!

Benefits of git

Git has benefits both for individuals and teams.

- Can resurrect any version, or go back and forth
- Allows multiple users to work in parallel

Downsides?

- A (sometimes frustrating) learning curve
- Some extra effort is required since synchronization is not automatic

Install git

You may already have git installed:

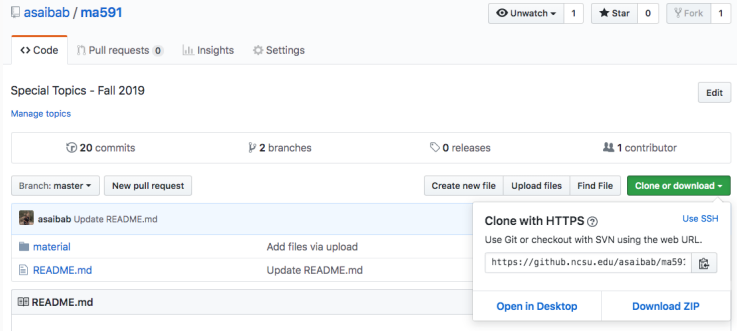
```
$ git --version  
git version 2.17.2 (Apple Git-113)
```

Many options to install available here:

- <https://www.atlassian.com/git/tutorials/install-git>
- Configure your username/email address after installation

Creating a repository

- Create an account on `https://github.ncsu.edu/`
- Can either clone or download this repository



In your folder of choice, type: `git clone <path to repo>`

- Check the status using `git status`

Typical Usage

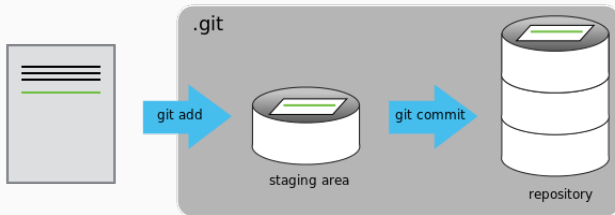
These is a typical workflow while using git

```
$ git pull  
$ git add <file or filenames>  
$ git commit -m "A descriptive message"  
$ git push
```

If you're the only person on a repository, these four commands may be sufficient.

Staging area

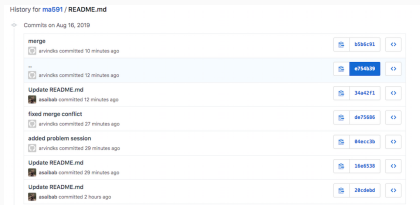
- `git pull` syncs your local copy to the repository. Don't forget this step!



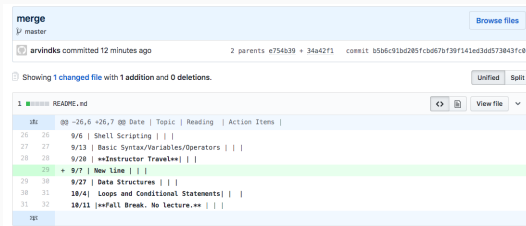
- Make all the changes to your heart's content
- `git add ...` adds the specific files to the staging area.
- `git commit ...` prepares it to be committed. So far, the files are not in the main repo yet!
- `git push` finalizes the commit.

History

Git stores the history of all your files



as well as the changes in each file



Conflicts

Conflicts

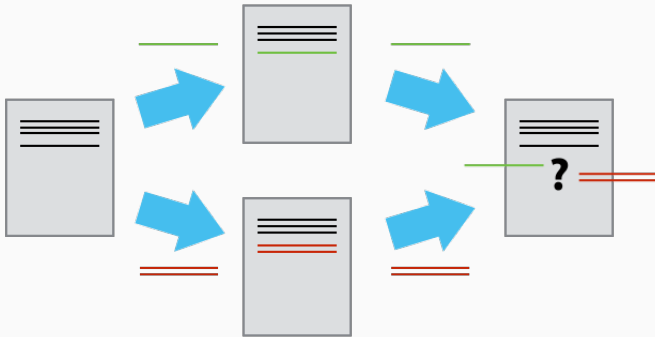
Scenario

- You have checked out a repository
- You make some local changes which you have not yet committed
- Your collaborator has pushed changes in the meantime, before you can commit your changes!

When it comes time to `git push`, you get an error like

```
[master 04ecc3b] added problem session
1 file changed, 1 insertion(+), 1 deletion(-)
(base) MA-C02V614GHTDG:ma591 arvindr$ git push
To https://github.ncsu.edu/asaibab/ma591.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.ncsu.edu/asaibab/ma591.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
(base) MA-C02V614GHTDG:ma591 arvindr$ _
```

Conflict: what to do?



Two cases:

- Edits happened in different files/different parts of files
 - `git pull` may merge the files automatically. An additional commit/push maybe necessary.
- Edits happened in the same part of file - Next slide.

Fixing conflicts in same part of file

When we do `git pull` again, the conflicted files have extraneous lines marking both versions of the conflict.

```
<<<<<< HEAD
12/6| In-class problem session | | |
=====
12/6| In-class lab session | | |
>>>>>> 16e6538e46cfb5fcccc4dca0fcba89396557c082
```

Steps:

- Delete extraneous lines and make the necessary changes.
- Do the add/commit/push routine.

When all else fails: the nuclear option!

The Ctrl-Alt-Del of git:

- Step 1: Delete your local copy (the entire folder)
- Step 2: Get a fresh copy from the website.

When all else fails: the nuclear option!

The Ctrl-Alt-Del of git:

- Step 1: Delete your local copy (the entire folder)
- Step 2: Get a fresh copy from the website.

Confession: I have done this several times!

Some tips

Removing files

- Use `git rm` (followed by `commit/push`) not `rm`.
- `rm` only removes the local copy. There is still a copy in the repository.

Always start your session with `git pull`!

- Saves a lot of work involving merge/conflict resolution¹

Don't upload large files.

- Upload speeds are slow. Find a different alternative, e.g., zenodo.

Miscellaneous

Ignoring files

Problem:

- `'git add *'` adds all files in a directory.
- This is an issue because there may be (machine dependent) auxilliary files which create conflicts for users.
- Ex. LaTeX users may want to not upload aux, toc, log, bbl files and only tex, bib, and figures

Solution: Create a `.gitignore` file in the desired folder. It may contain these lines

```
*.aux
```

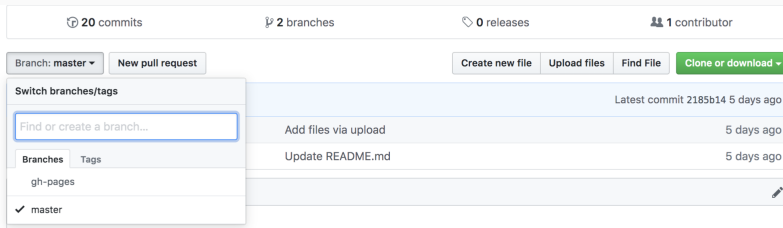
```
*.log
```

Useful tips:

- To override ignored files, use `git add -f *.aux`.
- The files to be ignored can be customized.

Branches

- Use a branch to isolate development work without affecting other branches in the repository.
 - Each repository has one default branch, and can have multiple other branches.
 - You can merge a branch into another branch using a pull request.
- You can use branches to:
- Develop features, Fix bugs, Safely experiment with new ideas



Github pages and markdown

- Github provides an easy formatting tool that uses Markdown language.
- This is simple and easy to learn and syntax fits on a page:
`https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet` We will also use Markdown in Jupyter notebooks.
- This is useful, e.g., for creating a README file (see our class repository).
- Github pages lets you turn it into a website with a few clicks:
`https://pages.github.com/`.