Lecture 3 – Version Control

Learning Objectives:

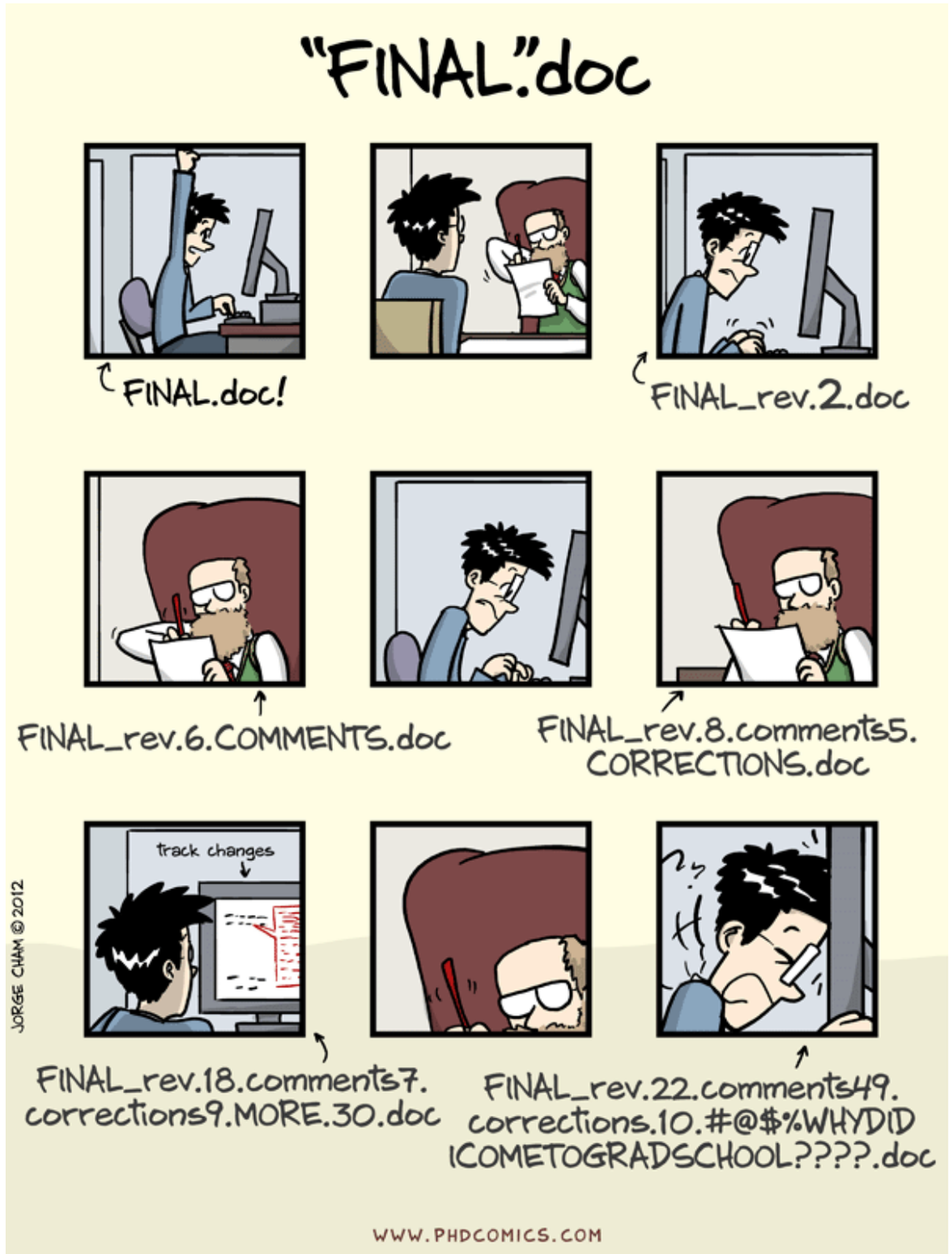
4. **Produce code that is reproducible and produces results that are replicable.**

    4.1 Learn commands of git.

    4.2 Learn how to explore history of a repository.

    4.3 Learn how to create and use remote repositories.

Why use version control?



http://www.phdcomics.com/

If you "break" your code, how do you get it back?

If your computer crashes, how do you get back your code?

If you update your code, how do you know which version you used for a particular analysis/graph?

If you work on multiple machines or between multiple people, how do you know what's going on?

Configure git:

Go to command line…

- MAC: Applications > Other > Terminal

- Windows: Start button > Search > “cmd”

Enter the following commands:

```
$  git config --global user.name githubusername
$  git config --global user.email youremail@chapman.edu
```

Assign your preferred command-line editor:

```
$  git config --global editor emacs
```

Start a git repository:

Make a new directory:

```
$  mkdir airplanes
```

Enter that directory:

```
$  cd airplanes
```

Initialize the repository:

```
$  git init
```

Create a file!

```
$  touch ilikeairplanes.txt
```

Add text to the file.

Basics of Operating git

Your computer

Saved!

```
Your code
New code
New code
```

You add some new code.

git add filename

Staged!

```
Your code
New code
New code
```

Changes to scripts recorded and catalogued.

git commit

Your git repository

```
* 8ffc40d - Tue, 1 Sep 2020 15:35:33 -0700 (25 minutes ago)
|          Updating link to Github in syllabus - lindsaywa
* d3b60a5 - Tue, 1 Sep 2020 14:33:23 -0700 (87 minutes ago)
|          Adding Lecture 02 Bash files - lindsaywaldrop
* abef621 - Mon, 31 Aug 2020 11:33:49 -0700 (28 hours ago)
|          Adding 01-Intro lecture slides - lindsaywaldrop
* 66ad4be - Thu, 27 Aug 2020 07:24:59 -0700 (5 days ago)
|          Adding coding standards - lindsaywaldrop
* de83348 - Wed, 26 Aug 2020 10:20:11 -0700 (6 days ago)
|          Updating readme - Lindsay Waldrop
* d135eed - Tue, 25 Aug 2020 10:57:47 -0700 (7 days ago)
|          Adding current Syllabus and Schedule - lindsayw
* 796cae6 - Tue, 25 Aug 2020 10:53:42 -0700 (7 days ago)
|          Updating readme files with additional instructi
* 0afa70a - Mon, 10 Aug 2020 15:23:26 -0700 (3 weeks ago)
           Initial commit - Lindsay Waldrop
CPSC-WALDROP-MBP:CourseInfoFall2020 waldrop$
```

Staged changes added to history

Check status of files in repository: `git status`

Move file back to unstaged: `git reset filename`

See differences since last commit:

```
git diff
git diff --staged
```

Remove a tracked file: `git rm filename`

Examining Repository History

Check your repository history: `git log`

commit id

History of a single file: `git log filename`

See differences between specific commits:

```
git diff HEAD~2
git diff HEAD~2 filename
git diff commitid
```

See a single file from a past commit: `git checkout commitid filename`

Revert unstaged changes: `git checkout -- filename`

Undo a commit: `git revert commitid`

Examining Repository History — Don't lose your HEAD! 👀

See the repository from a past commit: `git checkout commitid`

Entering 'Detached HEAD mode'!

```
* 8ffc40d - Tue, 1 Sep 2020 15:35:33 -0700 (86 minutes ago) (origin/master, origin/HEAD, master)
|          Updating link to Github in syllabus - lindsaywaldrop
* d3b60a5 - Tue, 1 Sep 2020 14:33:23 -0700 (2 hours ago)
|          Adding Lecture 02 Bash files - lindsaywaldrop
* abef621 - Mon, 31 Aug 2020 11:33:49 -0700 (29 hours ago)
|          Adding 01-Intro lecture slides - lindsaywaldrop
* 66ad4be - Thu, 27 Aug 2020 07:24:59 -0700 (5 days ago) (HEAD)
|          Adding coding standards - lindsaywaldrop
* de83348 - Wed, 26 Aug 2020 10:20:11 -0700 (6 days ago)
|          Updating readme - Lindsay Waldrop
* d135eed - Tue, 25 Aug 2020 10:57:47 -0700 (7 days ago)
|          Adding current Syllabus and Schedule - lindsaywaldrop
* 796cae6 - Tue, 25 Aug 2020 10:53:42 -0700 (7 days ago)
|          Updating readme files with additional instructions - Lindsay Waldrop
* 0afa70a - Mon, 10 Aug 2020 15:23:26 -0700 (3 weeks ago)
           Initial commit - Lindsay Waldrop
```

HEAD ⟵ Pointer to where you currently are in repository history

Master: original/default branch created when you started the repository

Origin: status of your remote repository (i.e. your stuff on Github)

Get back to most current version: `git checkout master`

```
* 8ffc40d - Tue, 1 Sep 2020 15:35:33 -0700 (89 minutes ago) (HEAD -> master, origin/master, origin/HEAD)
|          Updating link to Github in syllabus - lindsaywaldrop
```

Branches

- Branches are a way to create a new “copy” of your code to try different things, without affecting the main project.

  - Unlike copying the directory and working separately, these changes can be incorporated into the main project later by merging.

  - You can switch back and forth between your branches at will moving your HEAD.

- Create a branch by using **`git branch nameofbranch`**

```
CPSC-WALDROP-MBP:practicerepo waldrop$ git log
commit ecdcb0df3ff28c3d28fee5c71d4727e12bd68b1b (HEAD -> main, orig
in/main, origin/HEAD, twig)
Author: lindsaywaldrop <waldrop@chapman.edu>
Date:   Thu Aug 5 14:07:43 2021 -0700

    Adding a practice script and some info to README.

commit 9549728ac271d58d173f78bee90ab3dd8e517333
Author: Lindsay Waldrop <waldrop@chapman.edu>
Date:   Thu Aug 5 14:01:07 2021 -0700

    Initial commit
```

We’re still on the main branch. 👀

I named my new branch “twig”

Branches

- Switch to the new branch:

  `git checkout twig`

```
CPSC-WALDROP-MBP:practicerepo waldrop$ git log
commit ecdcb0df3ff28c3d28fee5c71d4727e12bd68b1b (HEAD -> twig, origin/main, origin/HEAD, main)
Author: lindsaywaldrop <waldrop@chapman.edu>
Date:   Thu Aug 5 14:07:43 2021 -0700

    Adding a practice script and some info to README.

commit 9549728ac271d58d173f78bee90ab3dd8e517333
Author: Lindsay Waldrop <waldrop@chapman.edu>
Date:   Thu Aug 5 14:01:07 2021 -0700

    Initial commit
```

Now we're on twig.

- Adding commits to each branch will cause a divergence. The two branches won't mix unless you merge them.

```
CPSC-WALDROP-MBP:practicerepo waldrop$ git hist
* ddb8e7e - Tue, 7 Sep 2021 14:31:22 -0700 (3 seconds ago) (HEAD -> main)
|           Adding a XKCD comic - lindsaywaldrop
| * 6a934f1 - Tue, 7 Sep 2021 14:28:50 -0700 (3 minutes ago) (twig)
|/            Adding some additional things - lindsaywaldrop
* ecdcb0d - Thu, 5 Aug 2021 14:07:43 -0700 (5 weeks ago) (origin/main, origin/HEAD)
|           Adding a practice script and some info to README. - lindsaywaldrop
* 9549728 - Thu, 5 Aug 2021 14:01:07 -0700 (5 weeks ago)
            Initial commit - Lindsay Waldrop
```

Branches

- Merge your branches with git merge:

```
git checkout main
git merge twig
```

```
[CPSC-WALDROP-MBP:practicerepo waldrop$ git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
[CPSC-WALDROP-MBP:practicerepo waldrop$ git merge twig
Merge made by the 'recursive' strategy.
 README.md                    | 4 +++-
 src/PracticeGithubScript.R | 6 +++++-
 2 files changed, 8 insertions(+), 2 deletions(-)
[CPSC-WALDROP-MBP:practicerepo waldrop$ git hist
*   512850d - Tue, 7 Sep 2021 14:36:46 -0700 (21 seconds ago) (HEAD
 -> main)
|\            Merge branch 'twig' - lindsaywaldrop
| * 6a934f1 - Tue, 7 Sep 2021 14:28:50 -0700 (8 minutes ago) (twig)
| |           Adding some additional things - lindsaywaldrop
* | ddb8e7e - Tue, 7 Sep 2021 14:31:22 -0700 (6 minutes ago)
|/            Adding a XKCD comic - lindsaywaldrop
* ecdcb0d - Thu, 5 Aug 2021 14:07:43 -0700 (5 weeks ago) (origin/ma
in, origin/HEAD)
|           Adding a practice script and some info to README. - lin
dsaywaldrop
* 9549728 - Thu, 5 Aug 2021 14:01:07 -0700 (5 weeks ago)
            Initial commit - Lindsay Waldrop
```

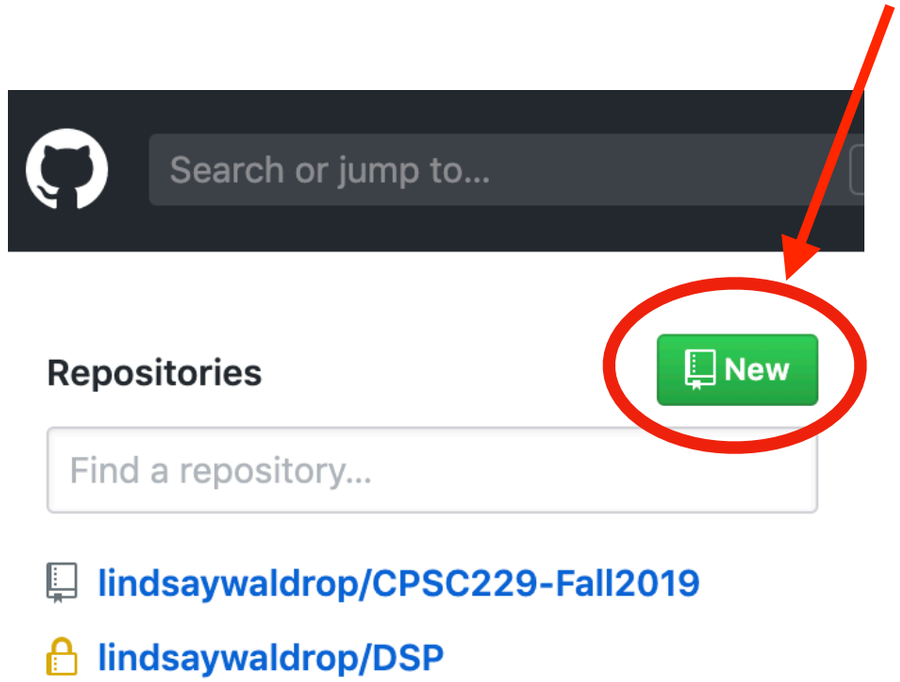
Stop here to practice!

Try adding several commits by adding some random files, or changing text in a file.

Move back and forth in your history to see the changes to your file space.

Start a New Repository on Github

1. Go to Github and in “Repositories” click the green “New Repository” button.



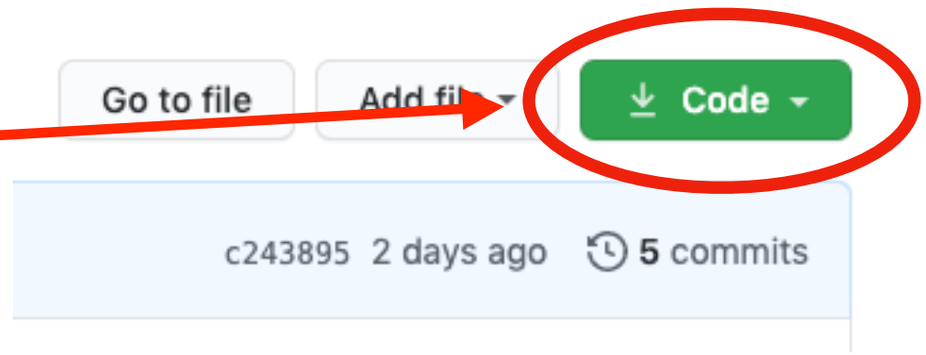
2. Pick a unique repository name (this will also be a folder name on your computer).

3. Click “Initialize this repository with a README file” then “Create Repository”

Cloning a Repository on Github

1. Go to the directory that you want the repository folder to be placed in.

2. Go to the repository's Github page and click "Code", copy the URL to your clipboard.



3. Clone the practice repository:

```
$  git clone COPIED_URL
```

4. Git will download the repository, create a folder, and place all of the code inside of that folder.

Basics of Operating git & Github

Your computer

Saved!

```
Your code
New code
New code
```

You add some new code.

git add filename

Staged!

```
Your code
New code
New code
```

Changes to scripts recorded and catalogued.

git commit

Your git repository

```
* 8ffc40d - Tue, 1 Sep 2020 15:35:33 -0700 (25 minutes ago)
|          Updating link to Github in syllabus - lindsaywa
* d3b60a5 - Tue, 1 Sep 2020 14:33:23 -0700 (87 minutes ago)
|          Adding Lecture 02 Bash files - lindsaywaldrop
* abef621 - Mon, 31 Aug 2020 11:33:49 -0700 (28 hours ago)
|          Adding 01-Intro lecture slides - lindsaywaldrop
* 66ad4be - Thu, 27 Aug 2020 07:24:59 -0700 (5 days ago)
|          Adding coding standards - lindsaywaldrop
* de83348 - Wed, 26 Aug 2020 10:20:11 -0700 (6 days ago)
|          Updating readme - Lindsay Waldrop
* d135eed - Tue, 25 Aug 2020 10:57:47 -0700 (7 days ago)
|          Adding current Syllabus and Schedule - lindsayw
* 796cae6 - Tue, 25 Aug 2020 10:53:42 -0700 (7 days ago)
|          Updating readme files with additional instructi
* 0afa70a - Mon, 10 Aug 2020 15:23:26 -0700 (3 weeks ago)
           Initial commit - Lindsay Waldrop
CPSC-WALDROP-MBP:CourseInfoFall2020 waldrop$
```

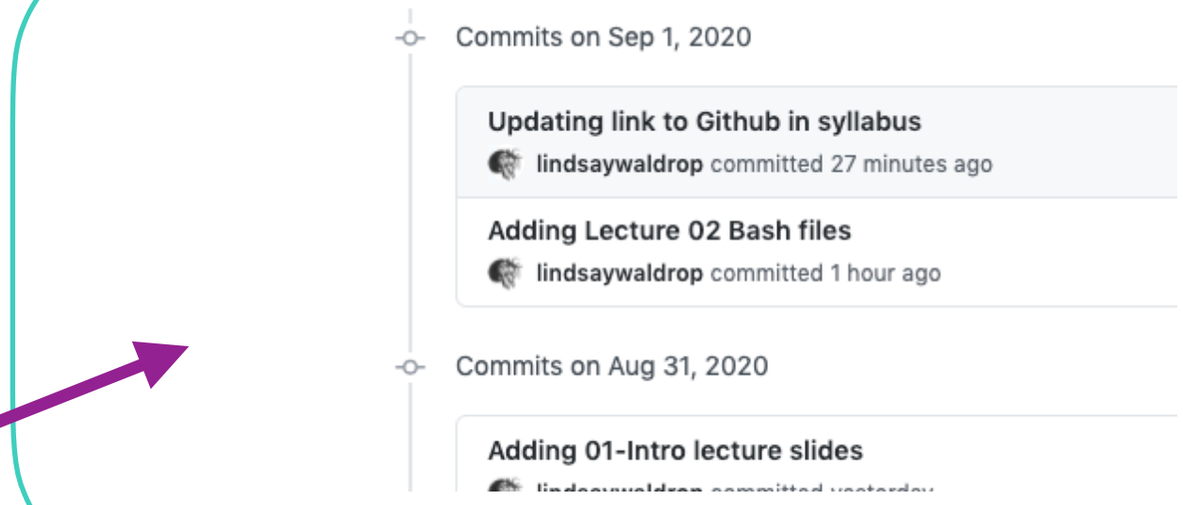
Staged changes added to history

PULL

PUSH

Github

Someone else's PUSH



Exploring a Cloned Github Project

- The cloned directory and the Github project will be identical to start out.

- As soon as a commit is made, they diverge and must be brought up-to-date with each other through **`push`** and **`pull`** commands.

- If two files are changed at the same time, you will need to merge your changes!

Merging commits (unconflicted)

- If the commits are **unconflicted** (no changes to the same file), the merge will be automatic when you push/pull.

  - git will prompt you to write a commit message about your merge:

  - Git will automatically merge:

Don't forget to push your changes to update Github!

```
CourseInfo — emacs ~/Dropbox (Chapman)/courses/CS510/Fall2021/CourseInfo/.git/MER...
File Edit Options Buffers Tools Help
Merge branch 'main' of https://github.com/CS510-Fall2021/CourseInfo
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.

Merging changes to schedule and README

CPSC-WALDROP-MBP:CourseInfo waldrop$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 656 bytes | 164.00 KiB/s, done.
From https://github.com/CS510-Fall2021/CourseInfo
   beff5ad..46b51e9  main       -> origin/main
hint: Pulling without specifying how to reconcile divergent branches is
hint: discouraged. You can squelch this message by running one of the following
hint: commands sometime before your next pull:
hint:
hint:   git config pull.rebase false  # merge (the default strategy)
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only       # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
Merge made by the 'recursive' strategy.
 README.md | 1 -
 1 file changed, 1 deletion(-)
```

You may want to set your strategy!

Merging commits (conflicted)

- If the commits are **conflicted** (changes to the same file), the merge will NOT be automatic when you push/pull. The process is a little more complicated, and requires you to manually review conflicts.

```
CPSC-WALDROP-MBP:CourseInfo waldrop$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 727 bytes | 242.00 KiB/s, done.
From https://github.com/CS510-Fall2021/CourseInfo
   df82cf1..ac6aa93  main       -> origin/main
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
CPSC-WALDROP-MBP:CourseInfo waldrop$
```

Merging commits (conflicted)

- First, **`git status`**:

```
CPSC-WALDROP-MBP:CourseInfo waldrop$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        LectureNotes/01-Intro/01-Intro-slides-presenter.pdf
        LectureNotes/02-Bash/02-LinuxBash.pdf

no changes added to commit (use "git add" and/or "git commit -a")
```

- **`git add README.md`**

- This will mark the lines that are different with each difference:

```
- After installing RStudio, make sure that git is working. Open RStudio and open\
 Preferences. Under the left column, select "Git/SVN" and make sure the "Enable \
version control interace for RStudio projects" is checked.

<<<<<<< HEAD
I'm making this change to demonstrated conflicted merging.
=======
This is a different line added, intended to create a merge conflict.
>>>>>>> ac6aa936b57dcd57efece33d4b95605d06bdfc9c
```

What's on your local repository

What's on Github

Merging commits (conflicted)

- Fix the line(s) manually, then save.
- You'll need to **`git add`** the file again

```
CPSC-WALDROP-MBP:CourseInfo waldrop$ git add README.md
CPSC-WALDROP-MBP:CourseInfo waldrop$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        #README.md#
        LectureNotes/01-Intro/01-Intro-slides-presenter.pdf
        LectureNotes/02-Bash/02-LinuxBash.pdf
```

- Commit the merge:

```
CPSC-WALDROP-MBP:CourseInfo waldrop$ git commit -m "Fixing merge issue with
E"
[main bbc3228] Fixing merge issue with README
```

- Finally, don't forget to push the fix!

```
CPSC-WALDROP-MBP:CourseInfo waldrop$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 647 bytes | 647.00 KiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To https://github.com/CS510-Fall2021/CourseInfo.git
   ac6aa93..bbc3228  main -> main
```

Forking: when you want your own repo

- Forking a repository is a way of copying all files into a new repository, which you own!

- Forking is great because you can run a completely independent repository, which can be **synced** with the original periodically to recover changes.

- You can also make a **pull request** on Github to suggest changes from your fork to the original repository.

- When should you clone and when should you fork?

  - **Cloning** is best if you are working directly on a project yourself or with your team that requires changes made to the same branch. Or, if you are not planning to make any changes at all to the code tracked in the directory.

  - **Forking** is best if you intend to make major changes to the project and would like ownership-level control. It is also best if you want to make changes to code that would not be great pushed to the original project.

Start a Repository from Command Line

1. Create a repository on your local machine and add at least one commit.

2. Go to Github and in “Repositories” click the green “New Repository” button.

3. Give your repository on Github THE EXACT SAME NAME as your existing repository (this will help sync them).

4. UNSELECT “Initialize this repository with a README file” then “Create Repository.”

5. Click the “Clone or download” button and copy the URL to your clipboard.

6. Return to to your shell and enter:

```
$  git remote add origin URLYOUCOPIED
$  git push --set-upstream origin master
```