

Taller Git

Carlos Enrique Jaramillo Aros

0075872

ceja0919@gmail.com

04 de agosto de 2016

1 ENCABEZADO

Differences Between Two Files

Enter the line numbers that have changed:

6,16,17,18

Old

```
1 <!doctype html>
2 <html>
3   <head>
4     <link rel="stylesheet"
5       type="text/css"
6       href="style-1.css">
7   </head>
8   <body>
9     <div id="content">
10    <h1 id="banner">Sunflowers</h1>
11    <div class="row">
12      <div class="image">
13        
14      </div>
15      <p class="description">
16        Some description here.
17
18
19    </p></div></div>
20  </body></html>
```

New

```
1 <!doctype html>
2 <html>
3   <head>
4     <link rel="stylesheet"
5       type="text/css"
6       href="style-!.css">
7   </head>
8   <body>
9     <div id="content">
10    <h1 id="banner">Sunflowers</h1>
11    <div class="row">
12      <div class="image">
13        
14      </div>
15      <p class="description">
16        Sunflowers were used to
17        remove toxins from a pond
18        after the Chernobyl disaster!
19    </p></div></div>
20  </body></html>
```

[VIEW INTRO](#)

[VIEW ANSWER](#)

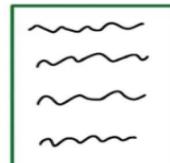
[SUBMIT ANSWER](#)

Finding Diffs Between Larger Files

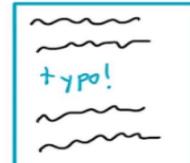
What word contains a typo in the new version of game.js but not in the old version?

space

Old version



New version



[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

When to Save

As a programmer, when would you want to have a version of your code saved?

- At regular intervals (e.g. every hour)
- Whenever a large enough change is made (e.g. 50 lines)
- Whenever there is a long pause in editing
- When you choose to save a version

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

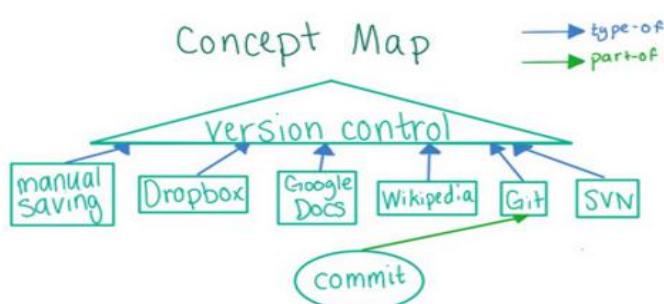
Concept Map: diff

Concept Map: diff

Remember that we are trying to avoid cluttering up our map, so try to only add relationships that would not be implied by other relationships already on the map.

If you were to put "git diff" on the map, which existing concept(s) would you directly connect it with?
Mark all that apply.

- commit
- Google Docs
- git
- Version Control

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

Would the following be good commit sizes?

Indicate whether you think each case would be a good size for a commit, too small (you'd like to wait and commit more changes together), or too big (you would have chosen to commit a smaller number of changes sooner).

Changes in Commit	Too Small	Just Right	Too Big
You commit all the changes required to add a new feature, which you've been working on for a week. You haven't committed since you started working on it.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
You find three typos in your README. You fix and commit the first.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
You commit all the changes required to add a new feature, which you've been working on for an hour.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
You fix two small bugs in different functions and commit them both at once.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

Tracking Across Multiple Files

Tracking across files
 Check the box next to each example below in which it makes sense to track multiple files together rather than separately:

- competition-style coding
- html and css files that make up a webpage
- photos you have photoshopped
- A novel, split up by chapter

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

Using git log and git diff

What is the ID of the commit with the message "Revert controls"?

b0678b161fcf74467ed3a63110557e3d6229cfa6

How many lines were added in that commit?

4

How many lines were deleted?

4

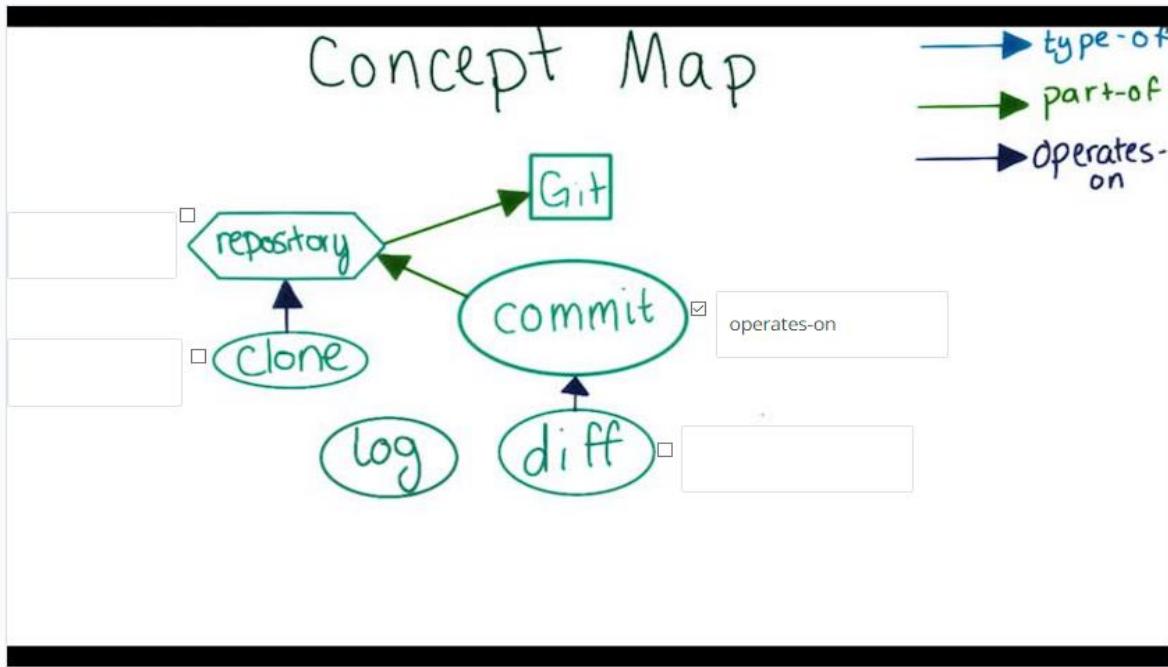
git diff first_id second_id

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Concept Map: repository, clone, log



[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Strange Warnings and Error Messages

Sometimes Git gives cryptic, confusing, or downright bizarre errors and warnings. In fact, we're about to run into one in the next exercise. Don't worry, we'll help you understand the message and what you can do about it. But just for fun, why don't you guess which of the following is not an actual message output by Git? You're not expected to know the answer to this, so just take a guess.

Which of the following is NOT a Git error or warning?

- Should not be doing an octopus
- You are in 'detached HEAD' state
- Panic! (the 'impossible' happened)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Checking Out Old Versions of Code

Finding the “buggy” commit

What is the ID of the first commit with the bug?

25ede836903881848fea811df5b687b59d962da3|



[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Survey Says!

So... What do you think?

Do you like the course so far? Have a suggestion? Share your opinions in this short survey: <http://goo.gl/Kygr2N>

- Click here when you're done



SUBMIT ANSWER

2 ENCABEZADO

Old File Plus Diff

Original File	Output of <code>diff -u</code>	Final File
A	---file.txt	
B	+++updated.txt	
C	B	
D	+\$	
E	C	
F	-D	
	+#	
	+%	
	E	

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Tracking Versions Using Git

Tracking versions using Git

Check all the statements that would be true if you tracked changes using Git:

- You would be able to see the difference between the two versions, but you would no longer be able to directly access the old version.
- Using `git diff` to compare the two versions would show the same changes as `diff -u` did in the previous exercise.
- The name of the file would change when you saved a second version in Git.
- To save two versions of the file, you would create two commits.

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Git command review

For each action, type the name of the corresponding Git command.

- Compare two commits, printing each line that is present in one commit but not the other.
- Make a copy of an entire Git repository, including the history, onto your own computer.
- Temporarily reset all files in a repository to their state at the time of a specific commit.
- Show the commits made in a repository, starting with the most recent.

[VIEW INTRO](#)
[VIEW ANSWER](#)
[SUBMIT ANSWER](#)

Behavior of git clone

Behavior of `git clone`

Select when each statement would be true.

	True when you copy	True when you clone	True in both cases
If someone else gives you the location of their directory or repository, you can copy or clone it to your own computer.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
The history of changes to the directory or repository is copied.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
If you make changes to the copied directory or cloned repository, the original will not change.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
The state of every file in the directory or repository is copied.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

[VIEW INTRO](#)
[VIEW ANSWER](#)
[SUBMIT ANSWER](#)

Behavior of git log

```
commit 7be5a12f1567866b0d77ccdf2055d1a33831da78
Date:   Fri Jul 11 12:56:26 2014 -0300

    Add sound for the wing.

commit 06d72e1f95f046002ec46f41cf71957227111141
Date:   Wed Jul 9 23:42:55 2014 -0300

    Add mute button.

commit 3d4d45b246aad6a1cd0afaf7cfae26966110727e
Date:   Mon Jul 7 17:35:47 2014 -0300

    Fix leaderboard button
```

For this `git log` output, check all the commits you would expect to contain code for a mute button.

- commit `7be5a12f1567866b0d77ccdf2055d1a33831da78` (the top commit listed)
- commit `06d72e1f95f046002ec46f41cf71957227111141` (the middle commit listed)
- commit `3d4d45b246aad6a1cd0afaf7cfae26966110727e` (the bottom commit listed)

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Behavior of git diff

```
commit 7be5a12f1567866b0d77ccdf2055d1a33831da78
Date:   Fri Jul 11 12:56:26 2014 -0300

    Add sound for the wing.

commit 06d72e1f95f046002ec46f41cf71957227111141
Date:   Wed Jul 9 23:42:55 2014 -0300

    Add mute button.

commit 3d4d45b246aad6a1cd0afaf7cfae26966110727e
Date:   Mon Jul 7 17:35:47 2014 -0300

    Fix leaderboard button
```

Select the command for which the mute button code lines would be shown as additions.

- `git diff 7be5a1 06d72e`
- `git diff 06d72e 7be5a1`
- `git diff 06d72e 3d4d45`
- `git diff 3d4d45 06d72e`

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Behavior of git checkout

Behavior of git checkout

Select when each statement would be true.

	Never true	Sometimes true	Always true
Checking out an earlier commit will change the state of at least one file.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Checking out an earlier commit will change the state of more than one file.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Checking out an earlier commit will change the state of every file in the repository.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
After checking out a commit, the state of all the files in the repository will be from the same point in time.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

Cloning a New Repository

Cloning a new repository

Clone the Pappu Pakia repository. The URL is given in the instructions and in the Instructor's Notes. Then open the game in your web browser. Which of the following people are creators of the game? Select all that apply.

- Kushagra
- Rishabh
- Shaurya
- Anaya

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

Buggy behavior

Both of these bugs were introduced by the same commit:

- There are no obstacles
- The pappu flickers across the screen

Which commit introduced the buggy behavior?

Copy and paste the commit ID, or enter the first four or more characters, in this box:

547f4171a82ec6429d002c1acef357aec41d3f17

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Fixing the Bug

Changes introduced

Which of the following changes was made? Check all that apply.

- An "or" expression was separated out into several "if" statements.
- Some variables were renamed.
- A function was refactored into two functions.
- Two functions were combined into one function.

File changed

Which file was affected by the buggy commit? Enter its name in this box:

js/utils.js

Obtain a working version of the code

Once you have a working version of the code, check this box:

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Second bug

Based on the commit messages, which commit do you think introduced the bug preventing the pappu clones from appearing?

Copy and paste the commit ID, or enter the first four or more characters, in this box:

```
003c8c197cd3b1e5b28b58f53ee14d7ebaa9bb3a
```

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

Fixing the Second Bug

Changes introduced

Which of the following changes was made in the buggy commit?

- In the buggy commit, the `x` and `y` coordinate of each clone is changed by a larger fixed amount.
- In the buggy commit, the `x` and `y` coordinate of each clone is changed by a larger random amount.
- In the buggy commit, the `velocity` of each clone is set to a larger fixed amount.
- In the buggy commit, the `velocity` of each clone is set to a larger random amount.

File changed

Which file was affected by the buggy commit? Enter its name in this box:

```
js/pappu.js
```

Obtain a working version of the code

Once you have a working version of the code, check this box:

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

3 ENCABEZADO

Initializing a Repository

New repository

How many commits do you think the new repository contains?

0

A commit is a snapshot of a Git repository

What command could you use to find out?

git log

[VIEW INTRO](#)

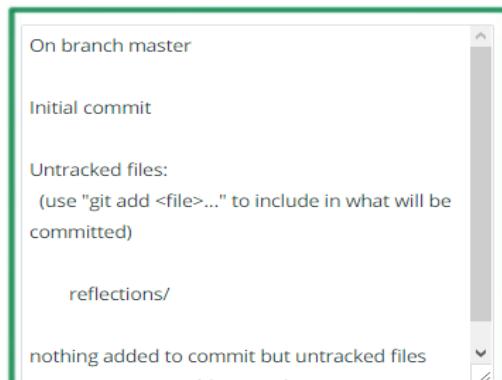
[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Examining the New Repository

Creating your reflections repository

Run the commands
git init and git status
in the directory with
your reflections.



```
On branch master
Initial commit
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    reflections/
nothing added to commit but untracked files present (use "git add" to track)
```

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Staging Area

Adding reflections to the staging area

Use git add to add whatever reflections files you would like to the staging area

Which file(s) will you add?
Why?

lesson_1_reflection.txt
because, its the file on my directory that i want to upload.

Paste the output of git status here:

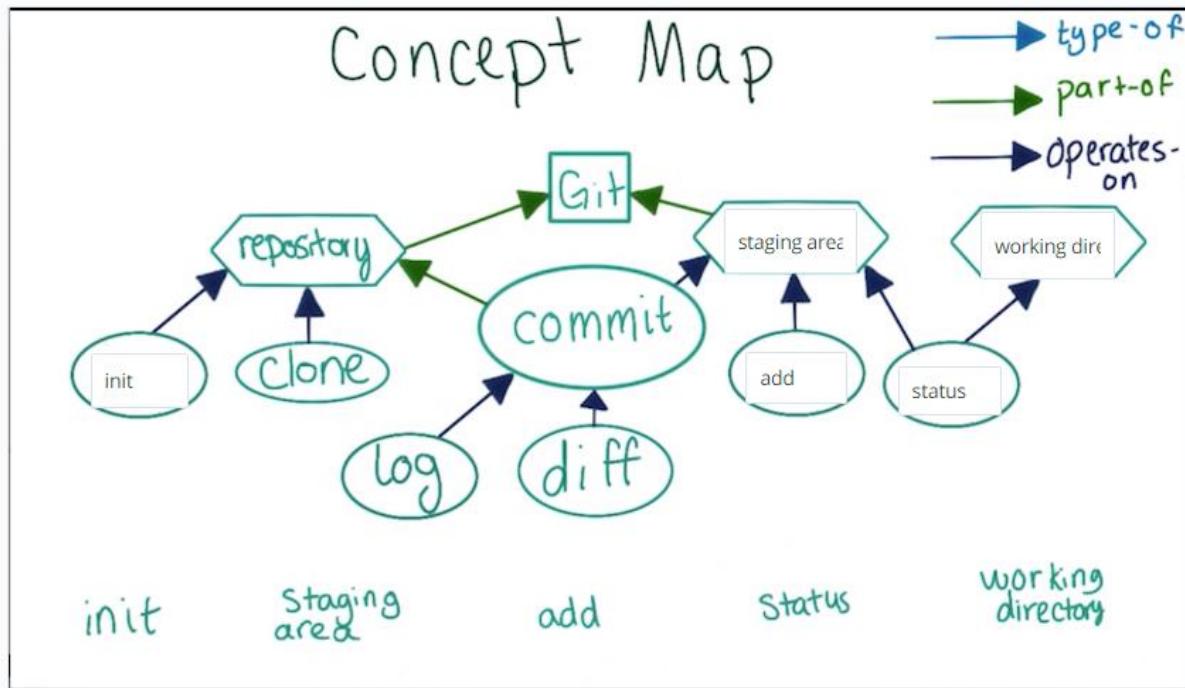
```
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
  
new file:  
lesson_1_reflection_prompts.txt
```

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Concept Map: init, add, staging area



[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

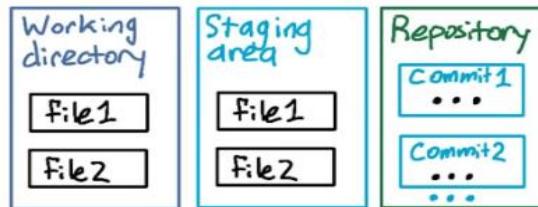
What two versions does each form of git diff compare?

Choices: working directory
staging area
commit 1
commit 2

git diff

git diff --staged

git diff commit1 commit2



[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Committing Changes

Commit changes in the staging area

- git commit

Commit all changes

- git add
- git status
- git commit

What is the output of git status after all your changes are committed?

nothing to commit, working tree clean

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Create an easy mode

- Create the easy-mode branch
- Checkout that branch
- Commit a change making the game easier
 - line 678

Paste the output of git status here:

```
On branch easy-mode
nothing to commit, working tree clean
```

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Branches for Collaboration

Commit history diagram

Draw a diagram of the recent commits in the asteroids repository, including the easy-mode, master, and coins branches.



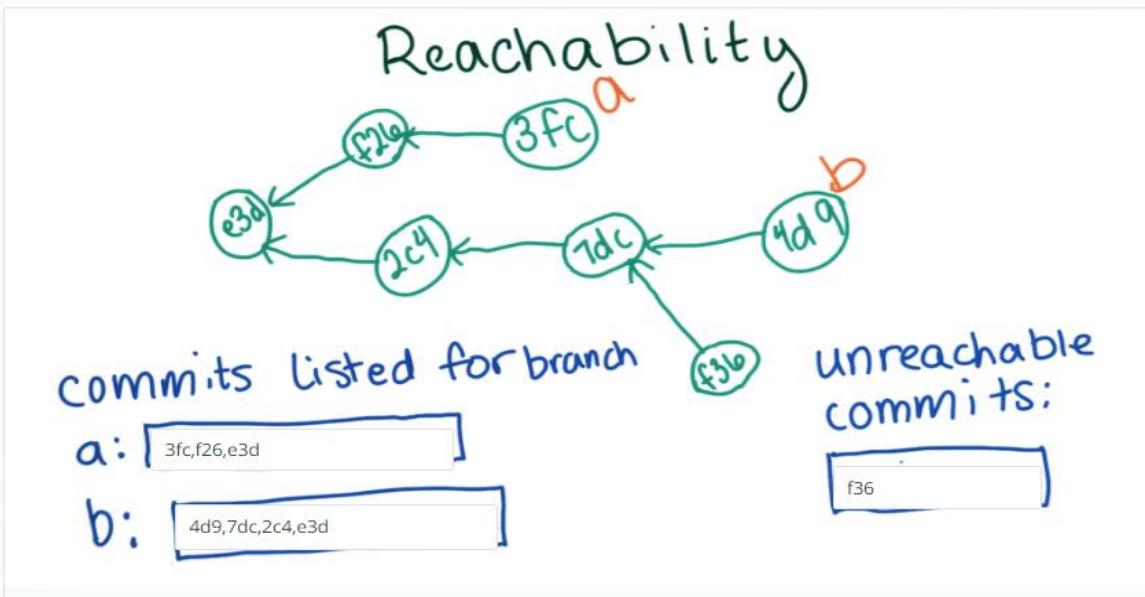
Try using git log to see what commits are on each branch.

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Reachability

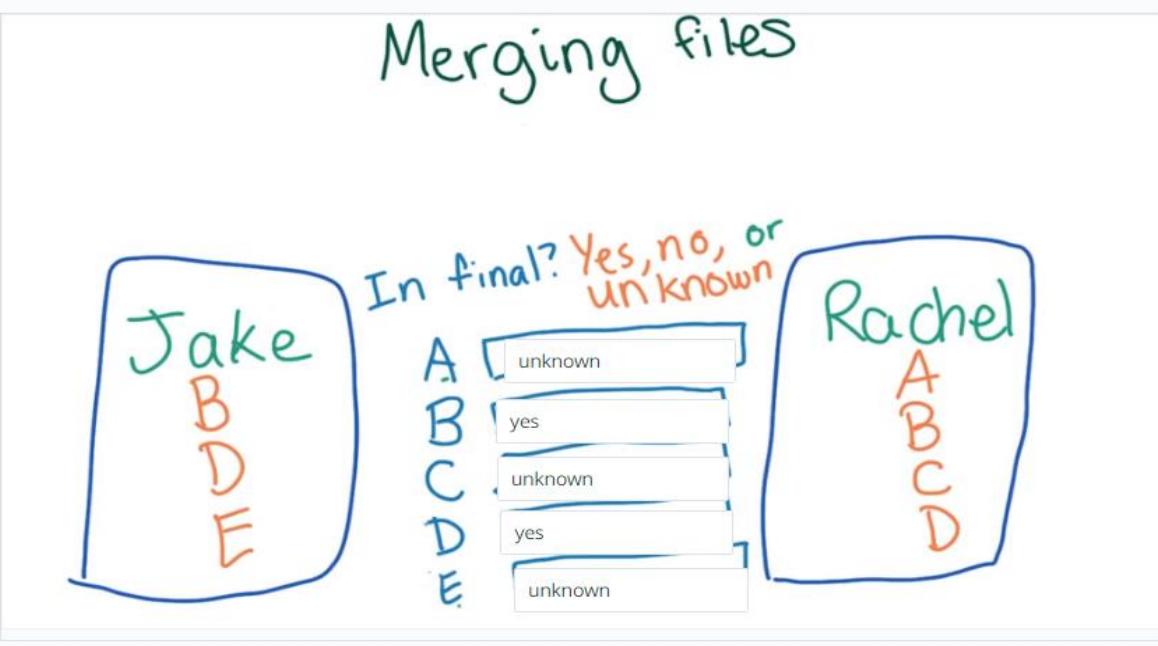


[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Combining Simple Files



[VIEW INTRO](#)

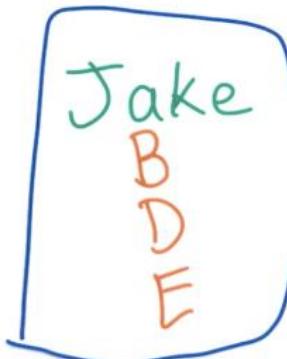
[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Merging files

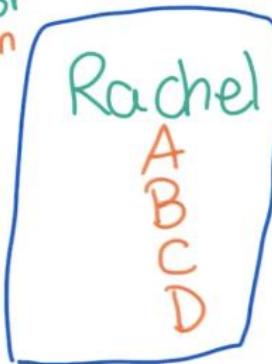
Original:

A
B
D



In final? Yes, no, or unknown

A	No
B	Yes
C	Yes
D	Yes
E	Yes



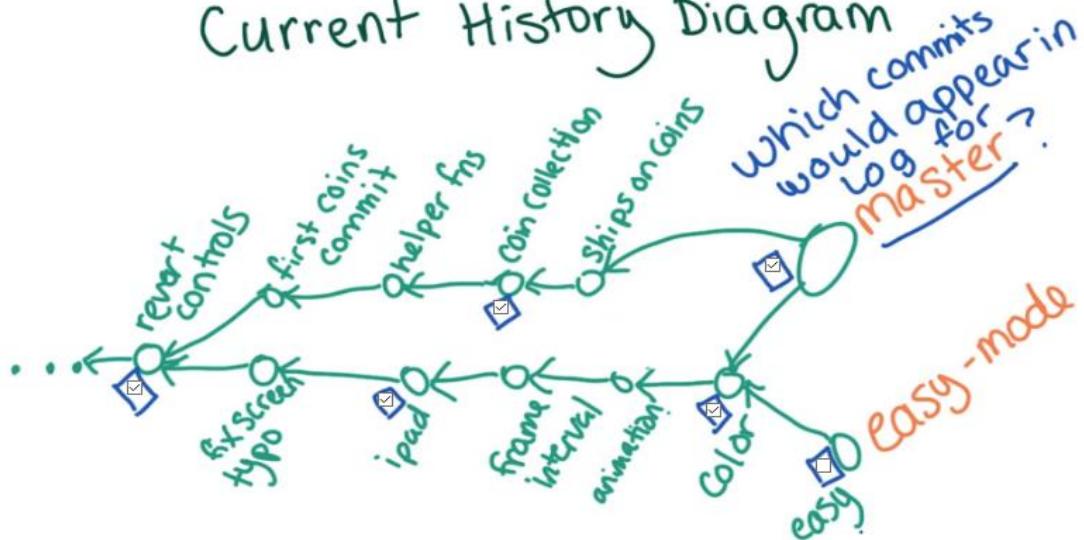
[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Merging Coins into Master

Current History Diagram



[VIEW INTRO](#)

[VIEW ANSWER](#)

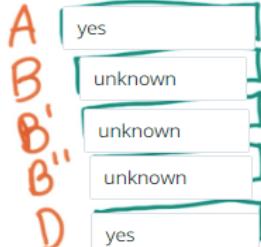
[SUBMIT ANSWER](#)

Merging files

Jake

A
B
D

In Final? yes, no, or unknown



Rachel

A
B
D

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Merging on the Command Line

Merging coins into master

- Repeat my steps on your own
- Optional: Make the coin solid and yellow

Did you merge the branches?

- Yes, and I made the coins yellow
- Yes, and I did not make the coins yellow
- No, I was not able to merge the branches
- No, I chose not to merge the branches

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

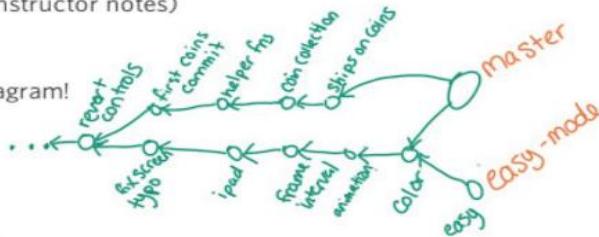
Update the diagram

Update the diagram to show what the state of the repository will look like after master is merged into easy mode.

Feel free to draw the diagram however feels most natural to you. A few options:

- use pencil and paper
- create a text file with stars and dashes similar to the output of `git log --graph`
- use an online diagramming tool (links in instructor notes)
- something we didn't think of!

Check here when you're done with the diagram!



[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Resolving Merge Conflicts

```

1- Asteroid = function() {
2-   this.breakIntoFragments = function () {
3-     for (var i = 0; i < 3; i++) {
4-       var roid = $.extend(true, {}, this);
5-       roid.vel.x = Math.random() * 6 - 3;
6-       roid.vel.y = Math.random() * 6 - 3;
7-       if (Math.random() > 0.5) {
8-         roid.points.reverse();
9-       }
10-      roid.vel.rot = Math.random() * 2 - 1;
11-      roid.move(roid.scale * 3); // give them a little push
12-      Game.sprites.push(roid);
13-    }
14-  };
15-
16-  this.collision = function (other) {
17-   SFX.explosion();
18-   if (other.name == "bullet") Game.score += 120 / this.scale;
19-   this.scale /= 3;
20-   if (this.scale > 0.5) {
21-     // break into fragments
22-     for (var i = 0; i < 2; i++) {
23-       var roid = $.extend(true, {}, this);
24-       roid.vel.x = Math.random() * 6 - 3;
25-       roid.vel.y = Math.random() * 6 - 3;
26-       if (Math.random() > 0.5) {
27-         roid.points.reverse();
28-       }
29-       roid.vel.rot = Math.random() * 2 - 1;
30-       roid.move(roid.scale * 3); // give them a little push

```

```

04 08 2016 14:06:23.489:
with unlimited concurrency
04 08 2016 14:06:23.654:

```

Resolving a merge conflict

Merge master into easy-mode on your computer

Paste the output of `git log -n 1` here:

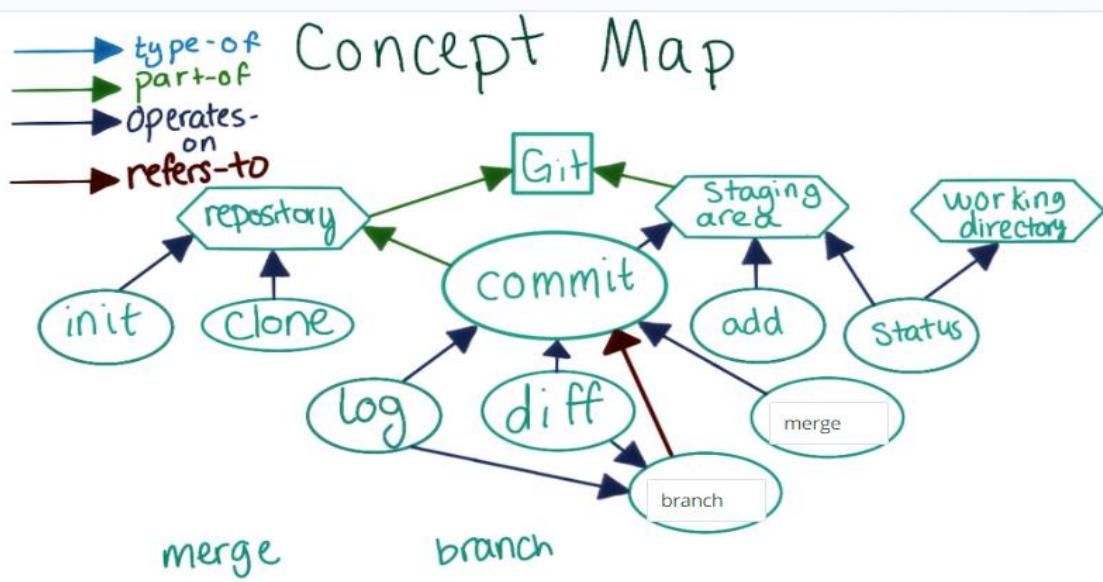
```
Author: CarlosJaramillo <ccea0919@gmail.com>
```

```
Date: Thu Aug 4 08:05:14 2016 -0500
```

```
Make asteroids split into 2 smaller pieces instead of 3
```

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

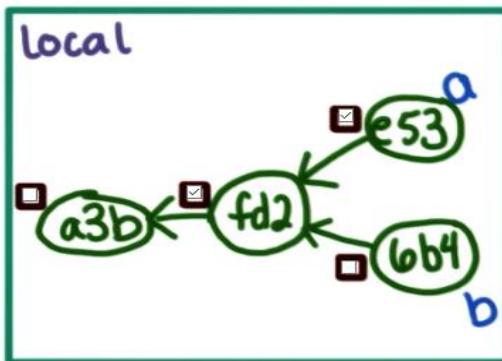
Concept Map: branch, merge

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

4 ENCABEZADO

Keeping Repositories in Sync

Syncing Repositories



Check the commits that will be sent

[VIEW INTRO](#)

[VIEW ANSWER](#)

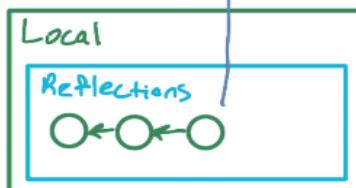
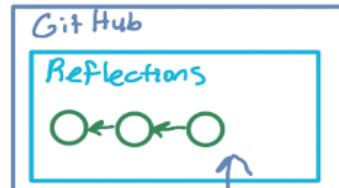
[SUBMIT ANSWER](#)

Adding a Remote

Sharing changes on GitHub

- Create a reflections repository on GitHub
- Push your reflections to GitHub
- Post a link to your reflections on the forums

Check here
when you're done:



[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Pulling changes from GitHub

Pull the changes you made on GitHub into your local repository

Which of the following is true ?

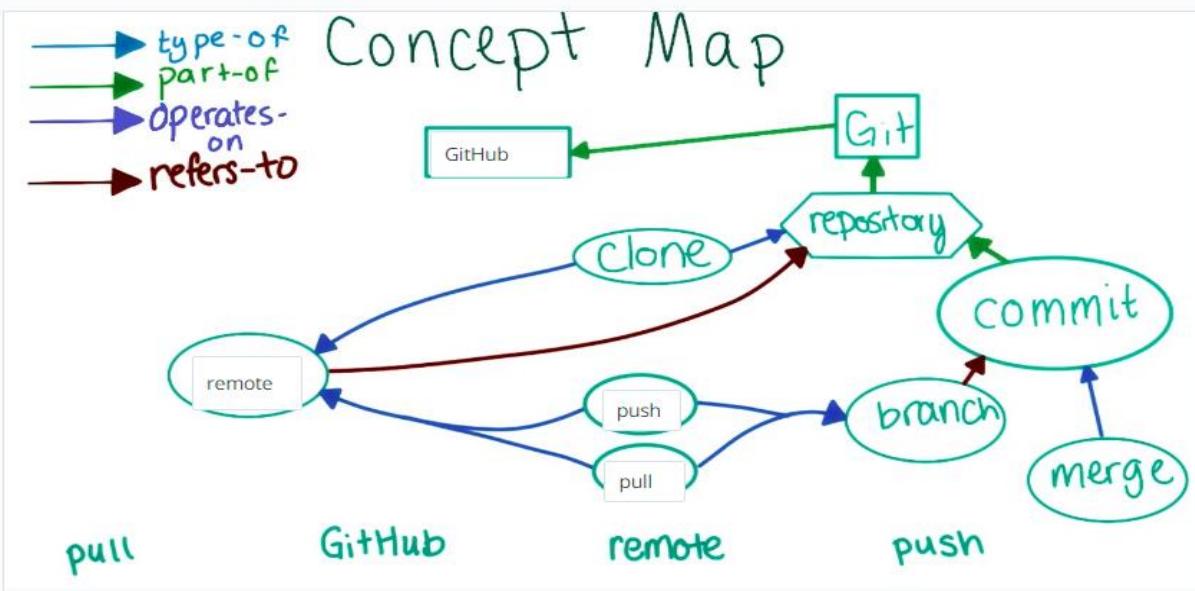
- Some changes in your local repository are not on GitHub
- Some changes on GitHub are not in your local repository
- Neither

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Concept Map: GitHub, Push, Pull, Remote

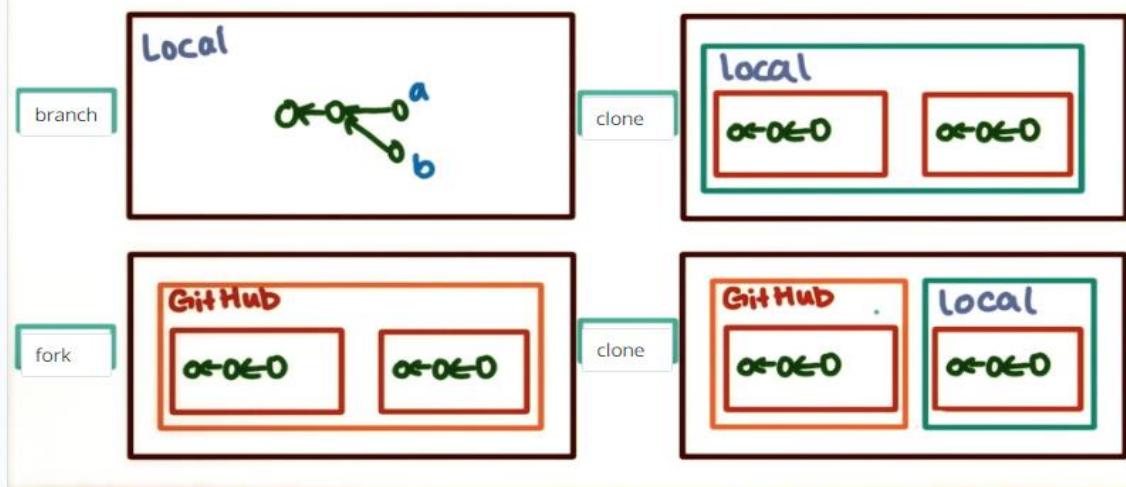


[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Clone, Branch, or Fork?

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

Push Changes to the Recipes Repository

Where was your commit present?

Right before you ran `git push`, where do you think your commit adding a new recipe was present? Similarly, now that you have run `git push`, where do you think this commit is present? Check all that apply, and remember that it's possible for a commit to be present in a repository both before and after running `git push`.

	Right before running <code>git push</code>	Right after running <code>git push</code>
In your local repository (visible via <code>git log</code>)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
On your fork (visible via the commit history on GitHub)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
On Larry's repository (visible via the commit history on GitHub)	<input type="checkbox"/>	<input type="checkbox"/>

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

Fork the Recipes Repository

Fork the recipes repository

- Fork Larry's repository
- Clone your fork
- Run `git remote -v`

```
origin https://github.com/carlosjara/recipes.git (fetch)  
origin https://github.com/carlosjara/recipes.git (push)
```

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Change the Chili Recipe

Change the chili recipe

To get conflicting changes, you'll want changes to be made by two people - you and Sarah. First, you should make your change locally. Modify the chili recipe by adding a new spice that you like to the ingredients list, and modify step 2 of the directions to mention that spice. Commit your change, but **don't push the change to your fork yet**. The following exercise will not work correctly if you push now.

Check this box when you're finished:

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Simulate Sarah's Changes

Where do these commits exist?

Once you've run the code to make it look like Sarah has modified your fork, run `git log` in the recipes repository on your computer, and look at the commit history of your fork on GitHub. For each commit, check whether it appears in your local repository or on your fork. (Note: some commits may exist in both places.)

	Exists locally	Exists on your fork
A commit by Larry with the message "Add a chili recipe"	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
A commit by you adding a new spice to the chili recipe	<input checked="" type="checkbox"/>	<input type="checkbox"/>
A commit by Sarah with the message "Remove cumin from chili"	<input type="checkbox"/>	<input checked="" type="checkbox"/>

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Updating Local Copies of Remote Branches

Output before and after `git fetch`

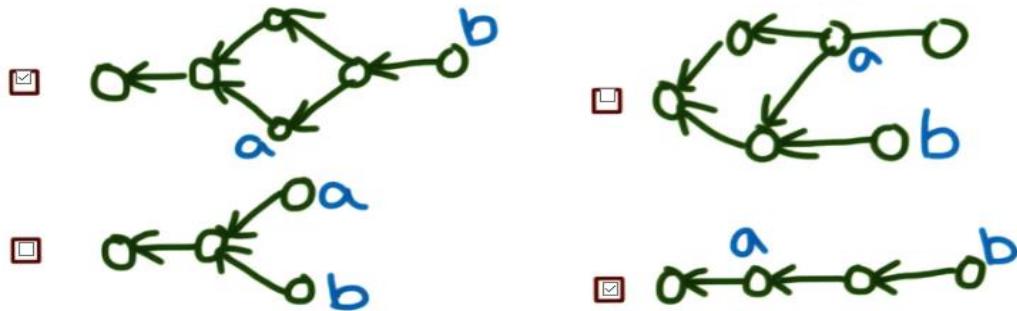
	git log origin/master	git status
Before git fetch	<input checked="" type="checkbox"/> Add chili recipe <input type="checkbox"/> New spice <input type="checkbox"/> Remove cumin	<input checked="" type="radio"/> Ahead <input type="radio"/> Behind <input type="radio"/> Up-to-date <input type="radio"/> Out-of-sync
After git fetch	<input checked="" type="checkbox"/> Add chili recipe <input type="checkbox"/> New spice <input checked="" type="checkbox"/> Remove cumin	<input checked="" type="radio"/> Ahead <input type="radio"/> Behind <input type="radio"/> Up-to-date <input type="radio"/> Out-of-sync

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Fast-forward Merges



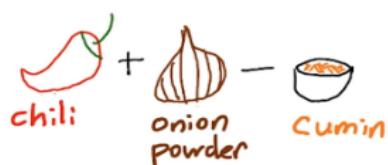
Check if merging **b** into **a** would be a fast-forward

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

Merging the Changes Together

Merging remote changes

- Create a merged version of your changes and Sarah's changes
- Push the merged version to GitHub
- Paste the output of `git status` here:

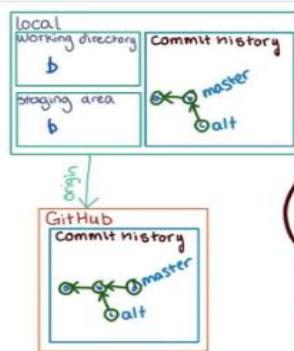


```
On branch master
Your branch is up-to-date with 'origin/master'

nothing to commit, working directory clean
```

[VIEW INTRO](#)[VIEW ANSWER](#)[SUBMIT ANSWER](#)

Making a Pull Request



What gets Changed?

	Local working directory	Local Staging area	Local master branch	Github master branch
edit and save README.md	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
git add README.md	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
git commit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
git pull	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
origin master	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
git push	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
origin master	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
merge alt pull request	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Updating a Pull Request

Make a pull request

- Create a different-oil branch
- Make a commit different-oil changing the oil
- Push different-oil to your fork
- Create a pull request from different-oil into master
 - Optional: Make another commit and update the pull request
- Paste a link to your pull request: <https://github.com/carlosjara/recipes/pull/197>

Don't merge the pull request

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Updating pull request to resolve conflicts

Repeat these steps yourself:

- Simulate Sarah's changes
- Update your different-oil branch with Sarah's changes
- Push the different-oil branch

The following should be true:

- Your local master contains Sarah's changes
- Your local different-oil contains the merge commit
- The pull request contains the merge commit
- The "merge" button is green

[VIEW INTRO](#)
[VIEW ANSWER](#)
[SUBMIT ANSWER](#)

Merging a Pull Request

If you completed the last exercise, then the "Merge" button on your pull request should be green. Now merge your pull request by pressing the button, then bring your local master up-to-date with the remote master using `git pull`.

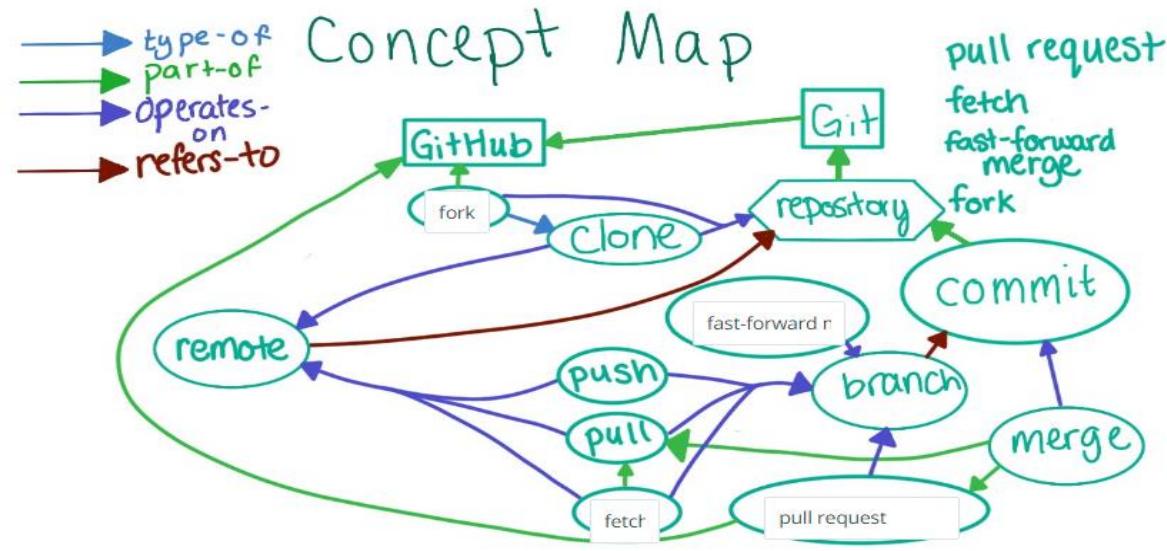
Once you're done, run `git log -n 1` on the master branch and paste the output in this box:

```
commit bc368511c6406028c77e2631f77c4d22a5da16d0
Merge: 79fff84 23d1775
Author: cbuckey
Date: Tue Sep 30 18:50:28 2014 -0400
```

Merge pull request #1 from cbuckey-uda/different-oil

Change vegetable oil to canola oil

[VIEW ANSWER](#)
[SUBMIT ANSWER](#)



[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

Keeping a Fork Up-To-Date

Updating a Cross-Repository Pull Request

Complete the following steps:

- Add the original repository as a remote in your clone
- Pull the master branch from the original repository
- Merge the master branch into your change branch
- Push your change branch to your fork

When you're finished, check here:

[VIEW INTRO](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)