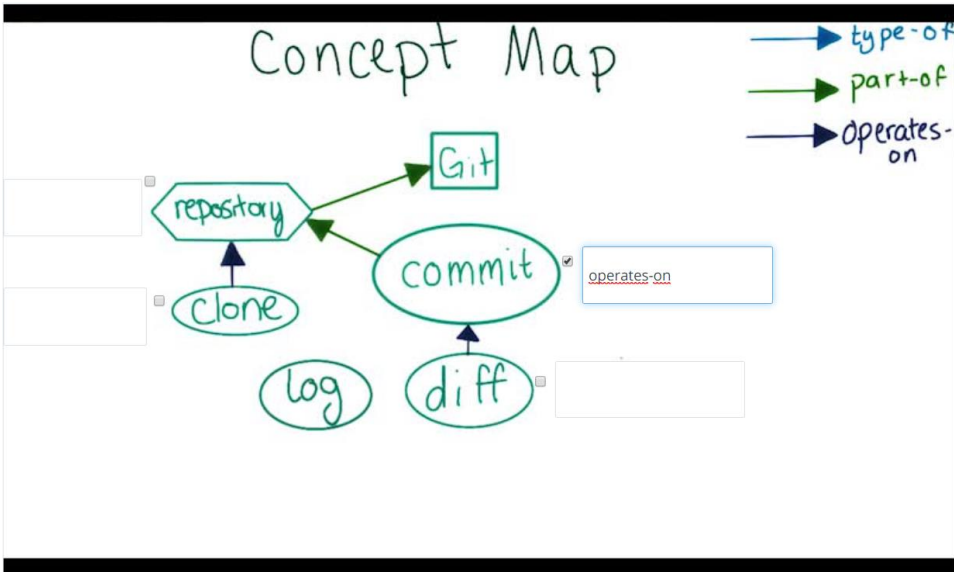


TAREA GITHUB

1.



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

2.

## 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.

3.

```
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)

4.

```
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`

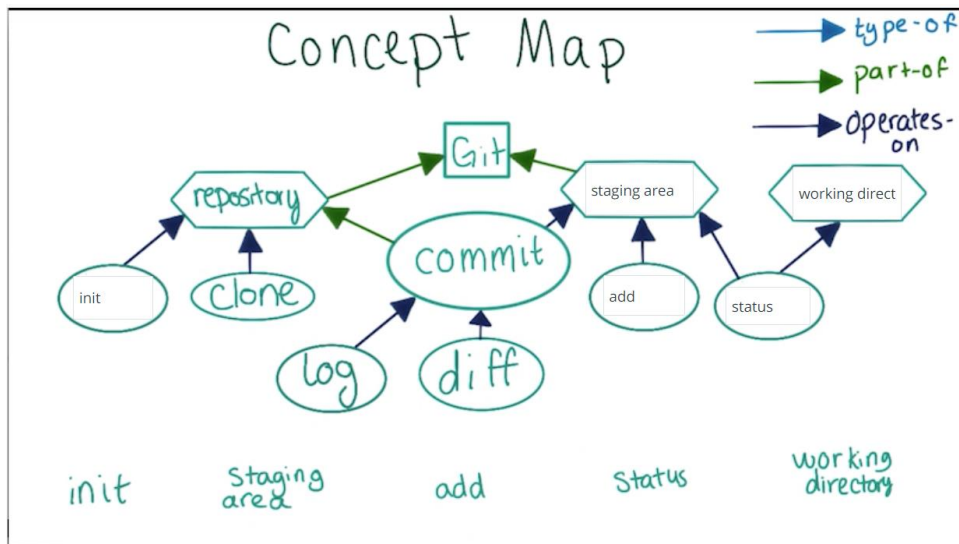
5.

### 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"/>

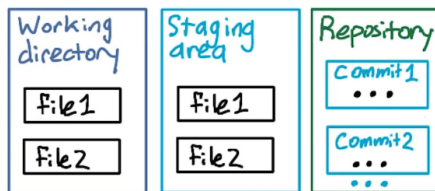
6.



7.

What two versions does each form of `git diff` compare?

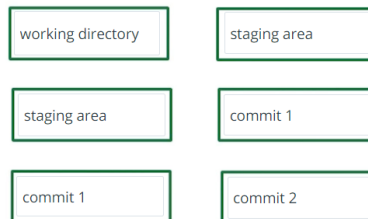
Choices: working directory  
staging area  
commit 1  
commit 2



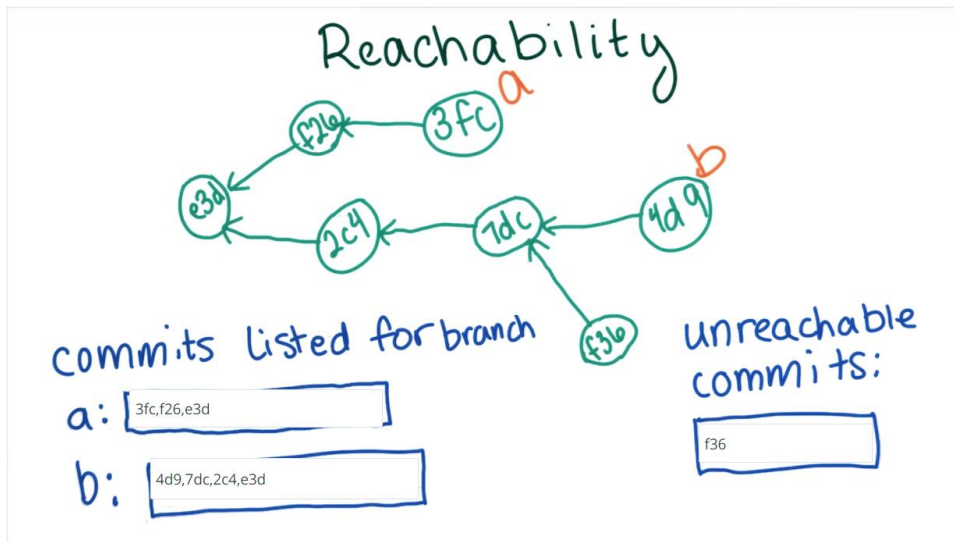
`git diff`

`git diff --staged`

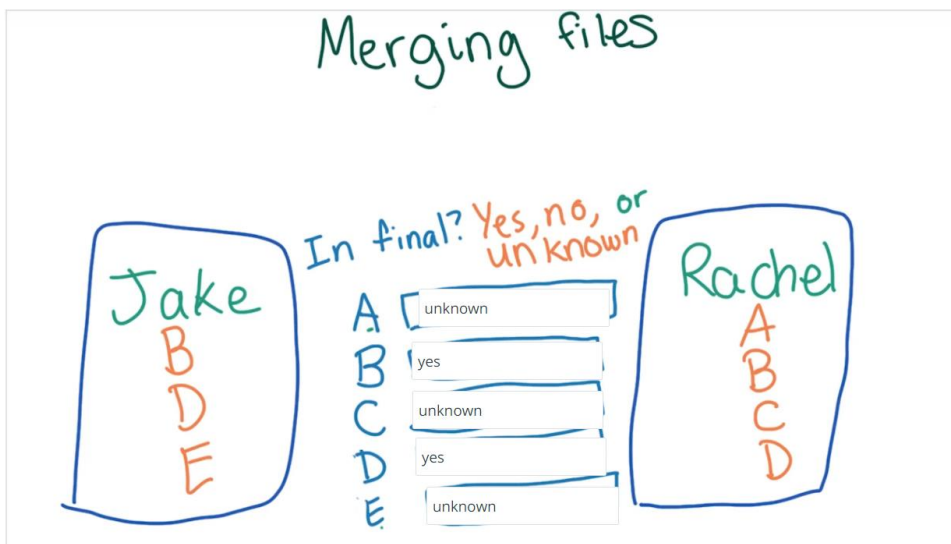
`git diff commit1 commit2`



8.



9.



10.

# Merging files

Original: A  
B  
D

In final? Yes, no, or unknown

Jake  
B  
D  
E

Rachel  
A  
B  
C  
D

A	<input type="text" value="no"/>
B	<input type="text" value="Yes"/>
C	<input type="text" value="yes"/>
D	<input type="text" value="Yes"/>
E	<input type="text" value="yes"/>

11.

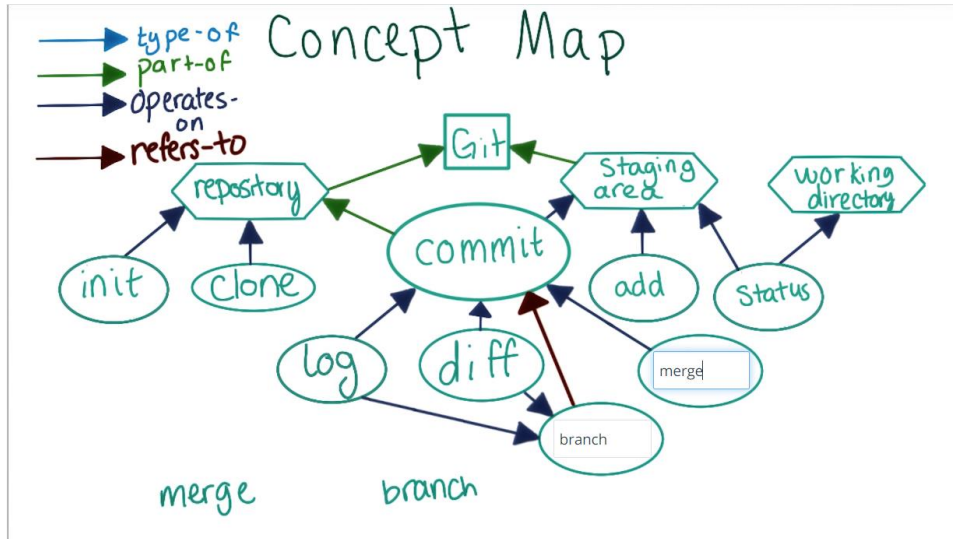
# Current History Diagram

The diagram illustrates a commit history graph. It features two main branches of commits. The bottom branch starts from a common ancestor (indicated by three dots) and includes commits for 'easy', 'Color', 'animation', 'frame interval', 'ipad', and 'fix screen type'. The top branch includes 'revert controls', 'first coins commit', 'helper fn', 'coin collection', and 'ships on coins'. A green oval highlights the 'Color' commit, with an arrow pointing to it from the text 'which commits would appear in log for master?'. Another arrow points from the 'ships on coins' commit to the 'Color' commit. The text 'easy-mode' is written in orange near the 'easy' commit.

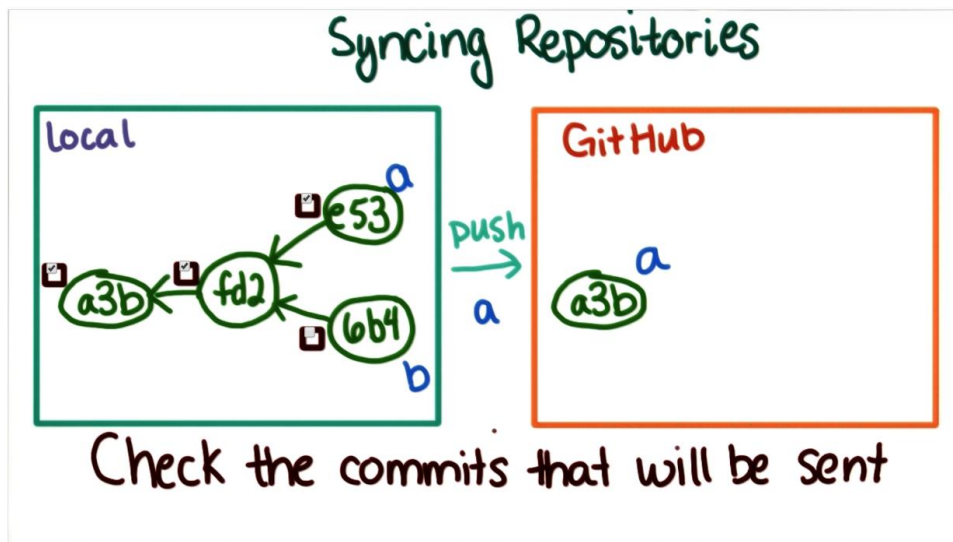
which commits would appear in log for master?

easy-mode

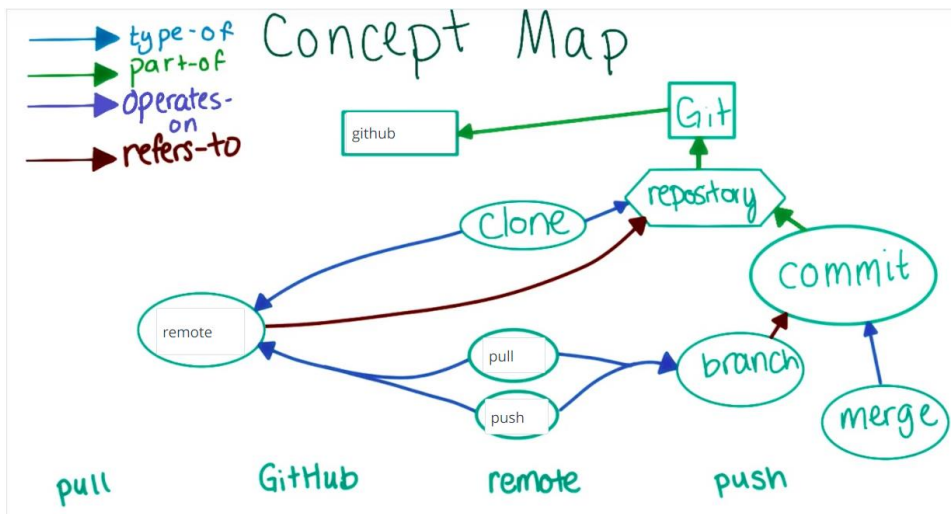
12.



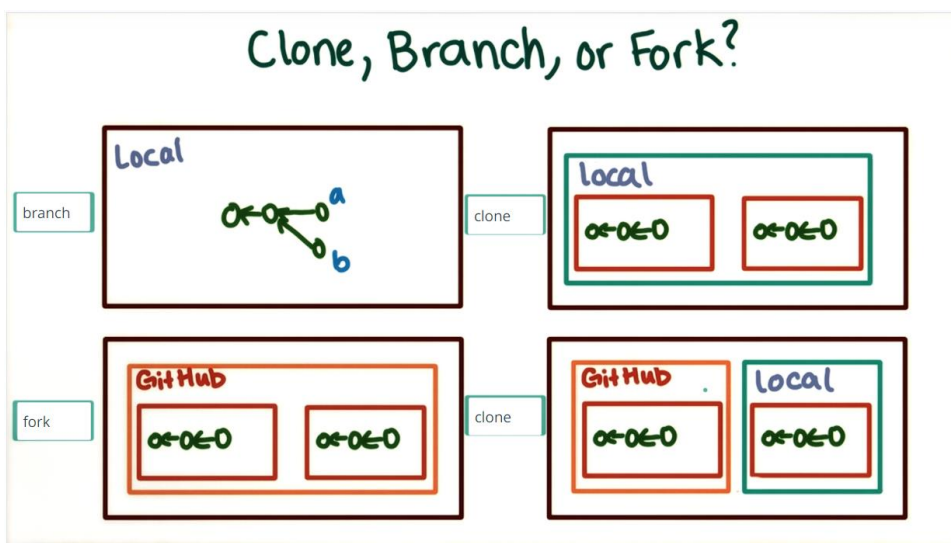
13.



14.



15.



16.



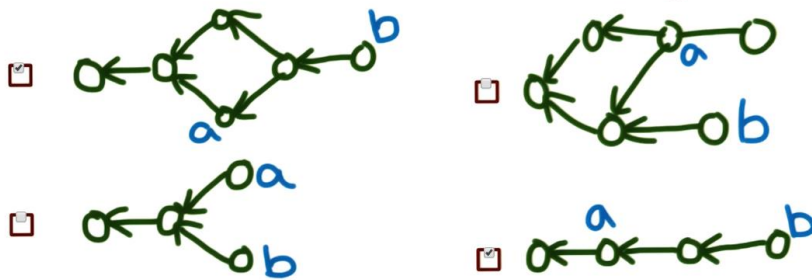
## Where was your commit present?

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"/>

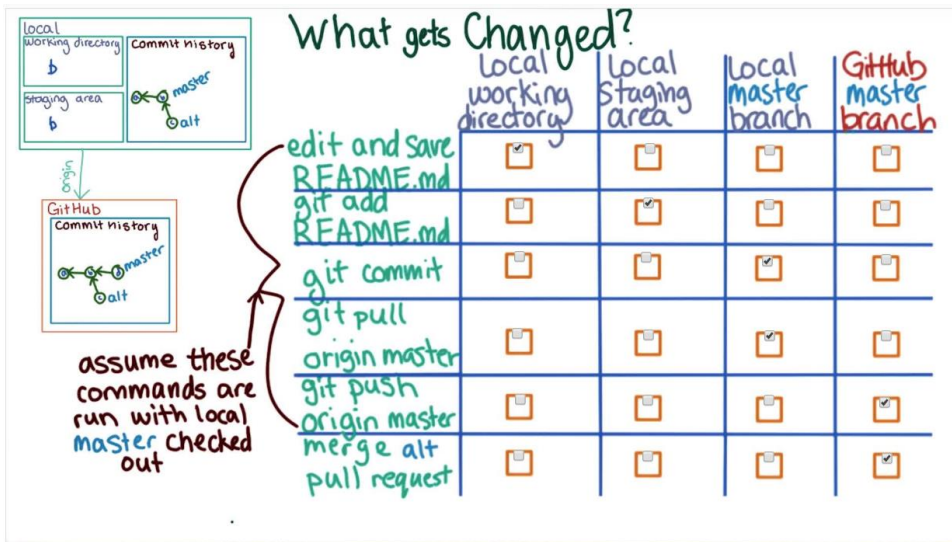
17.

## Fast-forward Merges

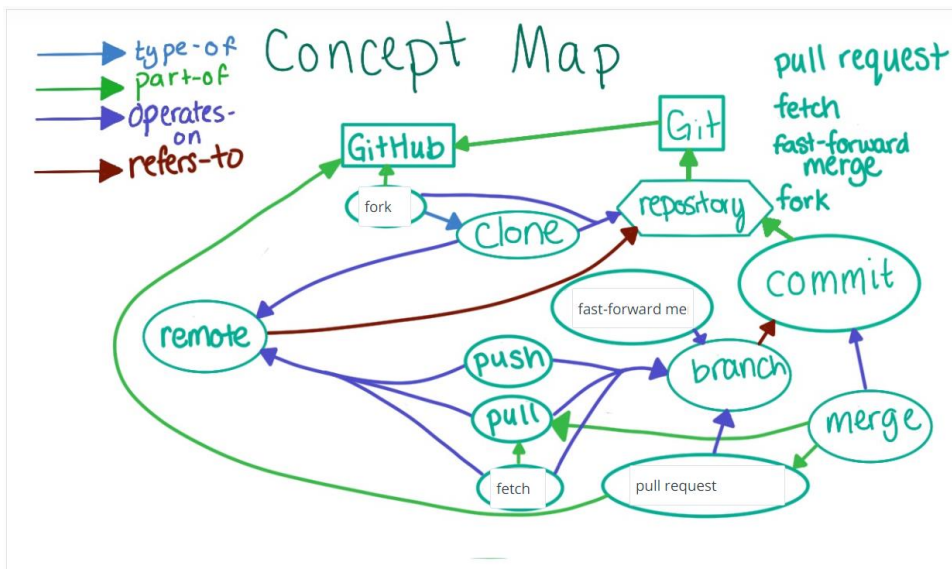


Check if merging **b** into **a** would be a fast-forward

18.



19.



20.