**Git Flow**

Objectives

Get familiar with git syntax

Get comfortable creating, merging, pushing, and pulling branches

Resolve merge conflicts

Instructions

Remember to fork and clone this lab if you haven't already.

Background

For this lab, you're going to be pretending that you work for National Geographic's Photography contest and that your assignment is to make a quick mockup of what the home page will look like. You want to include photos from the past five winners on the page.

**Branching**

Before altering your code base, open index.html in the browser. In the first part of this lab, you're going to add a photo of a turtle below the image of the tree to a branch then merge that branch into master.

Type git branch. This should show you that you're on the master branch.

Make a new branch called add-turtle from the master branch: git branch add-turtle

Type git branch again. Now you should see master (highlighted) and add-turtle

Switch to the add-turtle branch: git checkout add-turtle

Make sure you switched successfully by typing git branch again. This should return master and add-turtle (highlighted).

In index.html, below the tree picture and caption section, add the turtle picture using the HTML below:

<!-- begin turtle picture and caption -->

<div class="center-container">

<div class="card">

<div class="caption">

<h4>From the Sky</h4>

<p>Montse Grillo</p>

<p>Tenerife, Canary Islands</p>

</div>

</div>

<div class="card">

<img src="public/img/turtle.jpg" alt="sea turtle swimming">

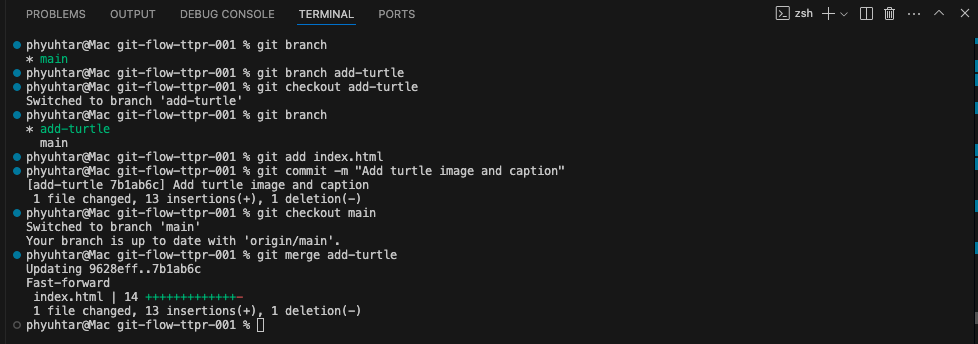
</div>

</div>

<!-- end turtle picture and caption -->

Open up your index.html and see how the page looks.

Stage and commit your changes.



**Merging**

Now that you've sucessfully added a turtle pic and caption to the add-turtle branch and staged and commited it, you're going to merge that branch into master.

The first step is to switch back to the master branch: git checkout master

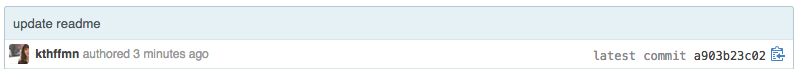
Now you're going to merge the add-turtle branch in: git merge add-turtle

Open up your index.html in the browser. How does it look? Does it have two pictures now?

While you have this change locally, your remote repo (the one up on Github) still thinks that index.html just has one picture, the tree/bird one. You need to tell it about this update.

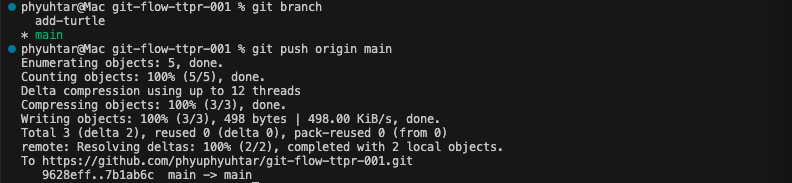
Push the update to your master branch on your remote repo: git push origin master

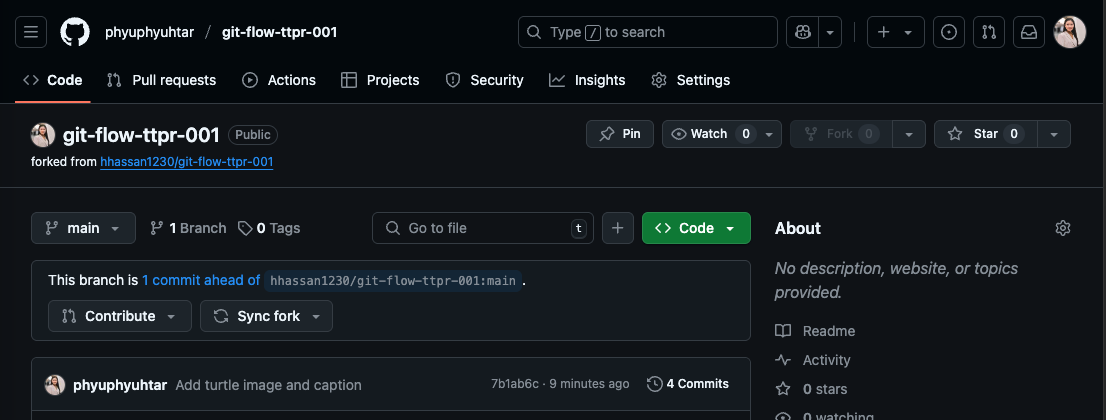
To make sure this push worked, visit your fork of this repo. From there, you can double check in at least two ways:

There will be a light blue bar above the file struture of the repo: [](https://camo.githubusercontent.com/b4b9ad1b1db614b695d4c50a65aef0824688d75a1fca2d4f02fc7e0c8f467068/68747470733a2f2f73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f726561646d652d70686f746f732f6769742d666c6f772d7570646174652d626c75652d6261722e706e67) that displays the most recent commit. This bar should have your GitHub picture followed by your GitHub name and a time stamp.

Click on the number of commits, it'll look something like this: [num of commits](https://camo.githubusercontent.com/1aaead57960b07024771878bb56adfa37b9d77bd6ddc74893e09803c08ae42a9/68747470733a2f2f73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f726561646d652d70686f746f732f6769742d666c6f772d7570646174652d636f6d6d6974732e706e67). The most recent commit, the one at the top, should be the one you made.

Let's get local again: How many branches do you expect to see when you type git branch? How many are there? What does this tell you about merging?





**Deleting A Local Branch**

Now that you've added the changes you've made from add-turtle to master, and master has been pushed to the remote repo successfully, it's time to delete your local version of add-turtle.

Make sure you're on master: git branch should return master highlighted.

Type git branch -d add-turtle

To make sure this branch was sucessfully deleted, type git branch. You should only see master now.



**Creating a Merge Conflict**

Walrus Branch

From master, make a new branch, add-walrus. On this branch, you're going to add the below code to index.html, under the tree and turtle pictures.

<!-- begin walrus picture and caption -->

<div class="center-container">

<div class="card">

<div class="caption">

<h4>Hello</h4>

<p>Misty Gage</p>

<p>Point Defiance Zoo, Tacoma, WA</p>

</div>

</div>

<div class="card">

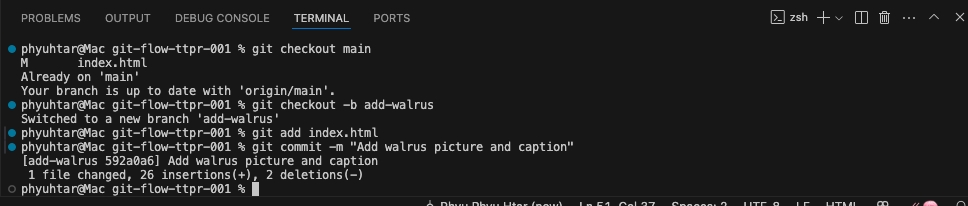
<img src="public/img/walrus.jpg" alt="walrus swimming with bubbles">

</div>

</div>

<!-- end walrus picture and caption -->

Remember to add and commit these changes.



Polar Bear and Walrus Branch

From the walrus branch, switch to master. From master, make a new branch, add-walrus-and-polar-bear. On this branch, you're going to add the below code to index.html, under the tree and turtle pictures (the same location where you added the walrus photo).

<!-- begin polar bear picture and caption -->

<div class="center-container">

<div class="card">

<img src="public/img/polar-bear.jpg" alt="walrus swimming with bubbles">

</div>

<div class="card">

<div class="caption">

<h4>Arctic Hi Five</h4>

<p>Colin Mackenzie</p>

<p>Svalbard, northern Norway</p>

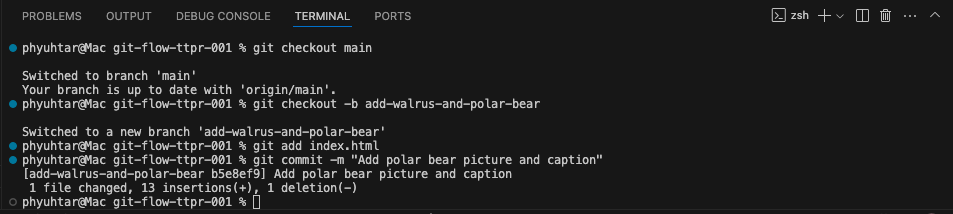
</div>

</div>

</div>

<!-- end polar bear picture and caption -->

Remember to add and commit these changes as well.



**Merging with Conflicts**

From the add-walrus-and-polar-bear branch, merge the add-walrus branch: git merge add-walrus

You'll probably see a message that looks like "Automatic merge failed; fix conflicts and then commit the result." This just means that you will have to open the files where there are merge conflicts, in this case index.html, and find the part that looks like:

<<<<<<< HEAD

HTML for polar bear

=======

HTML for walrus

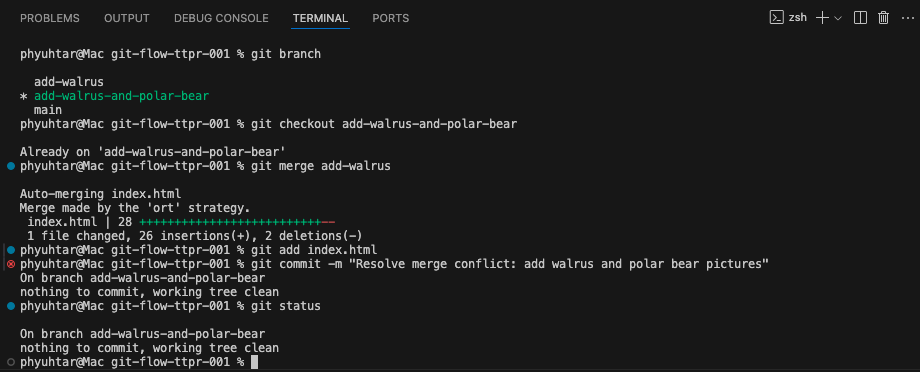
>>>>>>> add-walrus

Just decide which part of the markup you'd like to preserve: the first part from add-walrus-and-polar-bear, the second part from add-walrus, or both. Then delete the <<<<HEAD, ======, and >>>>> parts.

Fix the merge conflict in index.html so that index now has four photos: tree, turtle, walrus, and polar bear.

Remember, if you have multiple files with merge conflicts, you'll have to repeat this process with each file. Once you're done selecting which code to retain, git add and git commit these changes. Now when you type git status, your terminal should not display "You have unmerged paths."

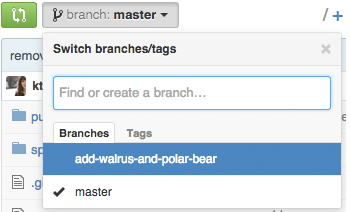
Add and commit these changes.



**Pushing a Local Branch**

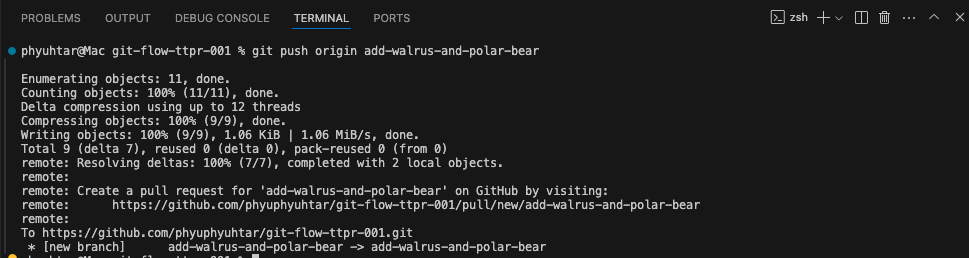
Now you're going to create a add-walrus-and-polar-bear branch on your remote repo.

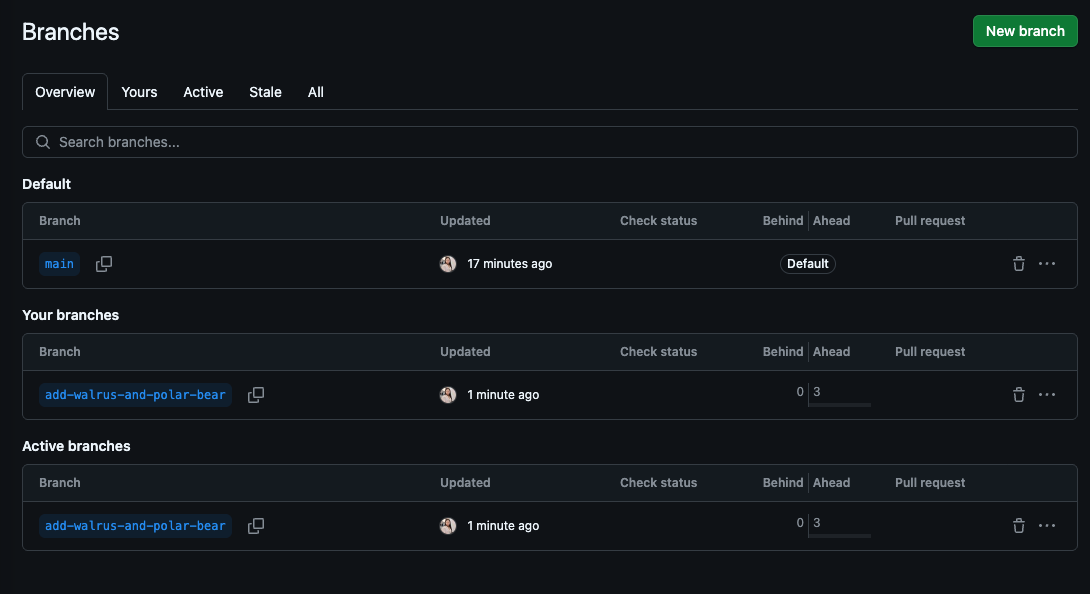
From the branch add-walrus-and-polar-bear, push the code to a remote branch of the same title. You can do this in one line with: git push origin add-walrus-and-polar-bear.

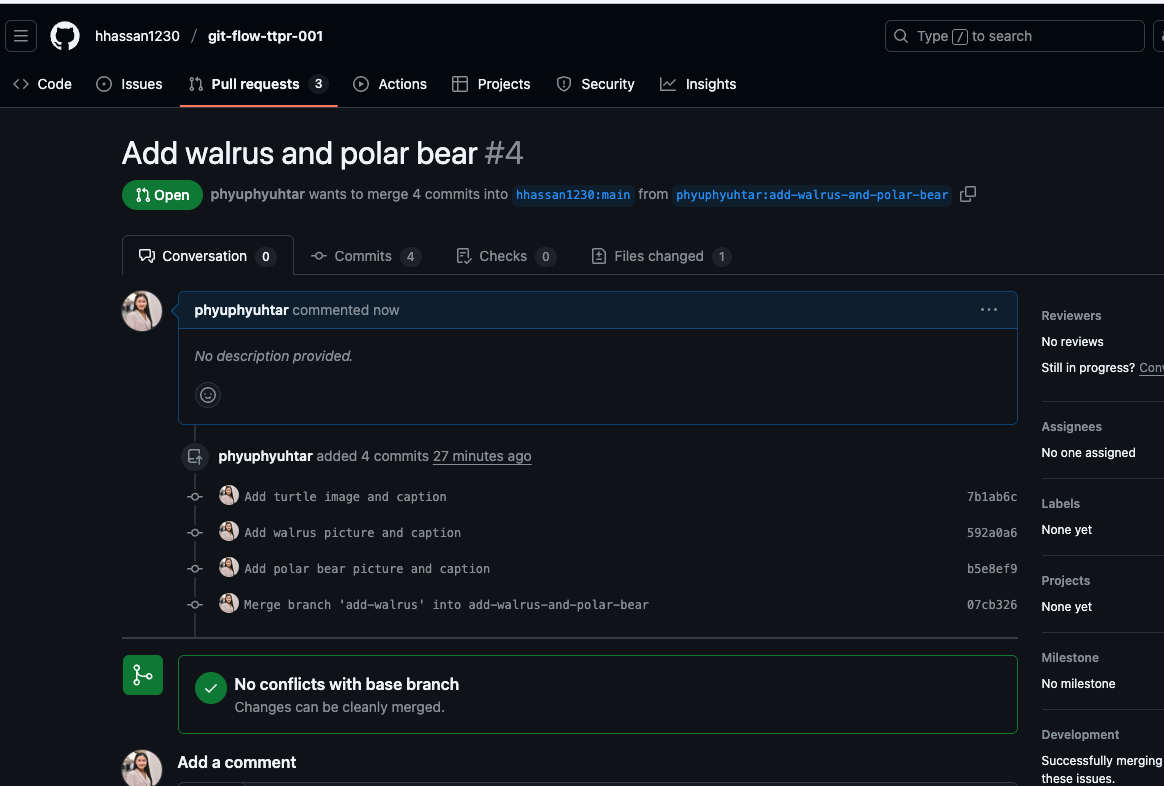
To ensure this push worked, head over to GitHub and view your forked repo. Click on the branch dropdown: [](https://camo.githubusercontent.com/2c689f34ea2d4749df789d4a929606ddc2abdecda41cfce22dd0cc282651ade9/68747470733a2f2f73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f726561646d652d70686f746f732f6769742d666c6f772d7570646174652d6272616e63682d64726f70646f776e2e706e67), there should be the option to view the add-walrus-and-polar-bear branch.

Now your master branch has a tree and a turtle while add-walrus-and-polar-bear has tree, walrus, and polar bear.

Since you merged add-walrus into add-walrus-and-polar-bear, go ahead and delete it. Remember that you cannot "be" on the branch that you're trying to delete so make sure you're on add-walrus-and-polar-bear instead.







**Getting a Remote Branch**

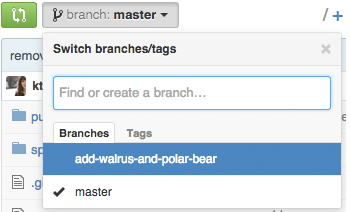
Many times when working in groups, a developer will branch off of master, add some code, then push this new branch to the remote repo for another developer to work on.

Since you're working on this project alone, you're going to mimic the remote creation of a new branch. For the next section, you're going to pretend to be a team member, Lauren. Lauren likes to add emoticons to readmes.

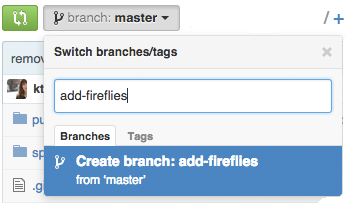
Lauren Time

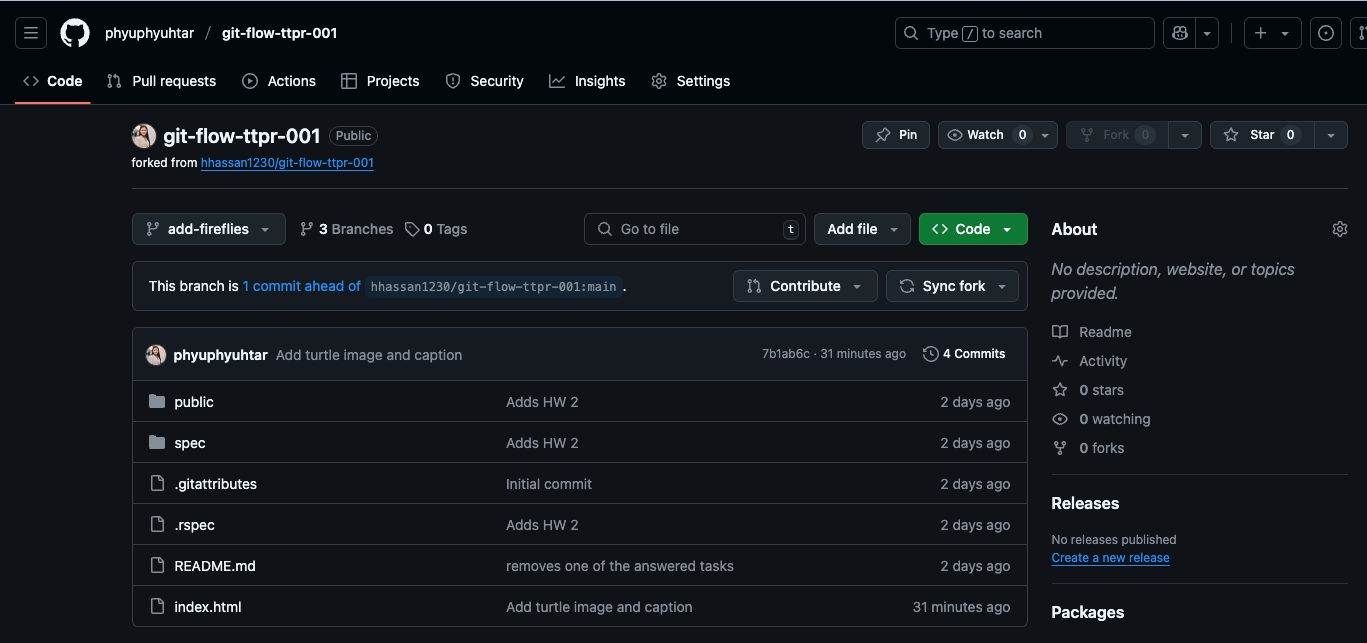
The first step is to pretend to be Lauren.

As Lauren, go to GitHub. From GitHub, go to your fork of this lab. From this interface, create a new branch called "add-fireflies" using the pictured interface below:

[](https://camo.githubusercontent.com/2c689f34ea2d4749df789d4a929606ddc2abdecda41cfce22dd0cc282651ade9/68747470733a2f2f73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f726561646d652d70686f746f732f6769742d666c6f772d7570646174652d6272616e63682d64726f70646f776e2e706e67)

Type "add-fireflies" then click on "Create branch: add-fireflies from 'master'", as pictured below:

[](https://camo.githubusercontent.com/4a1e5b95296ba53f08d4c7b2580e77e8d9078374d3d08f90790959ba199f4ce8/68747470733a2f2f73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f726561646d652d70686f746f732f6769742d666c6f772d7570646174652d66697265666c792d6272616e63682e706e67)



This will redirect you to a newly created branch on your remote repo called "add-fireflies". From here, click on README.md:

[readme](https://camo.githubusercontent.com/02ac5c5026ba064968f292ac3a97261906be3627754075c7f3afe74afa062d72/68747470733a2f2f73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f726561646d652d70686f746f732f6769742d666c6f772d7570646174652d726561646d652d6c696e6b2e706e67)

Now click on the pencil icon, shown below:

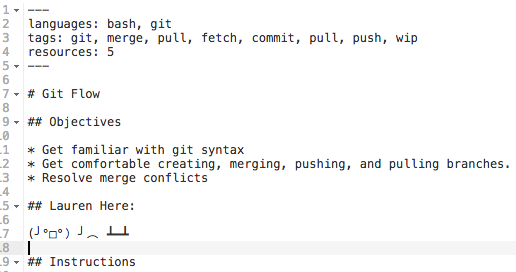
[edit icon](https://camo.githubusercontent.com/e07e81b8bbc206722dba2f7f88aa73aa9cee8f1180ff15d3fb5599afce1744b9/68747470733a2f2f73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f726561646d652d70686f746f732f6769742d666c6f772d7570646174652d656469742e706e67)

Near the top of the markup, below the "Objectives" section, add the text below:

## Lauren Here:

(╯°□°）╯︵ ┻━┻

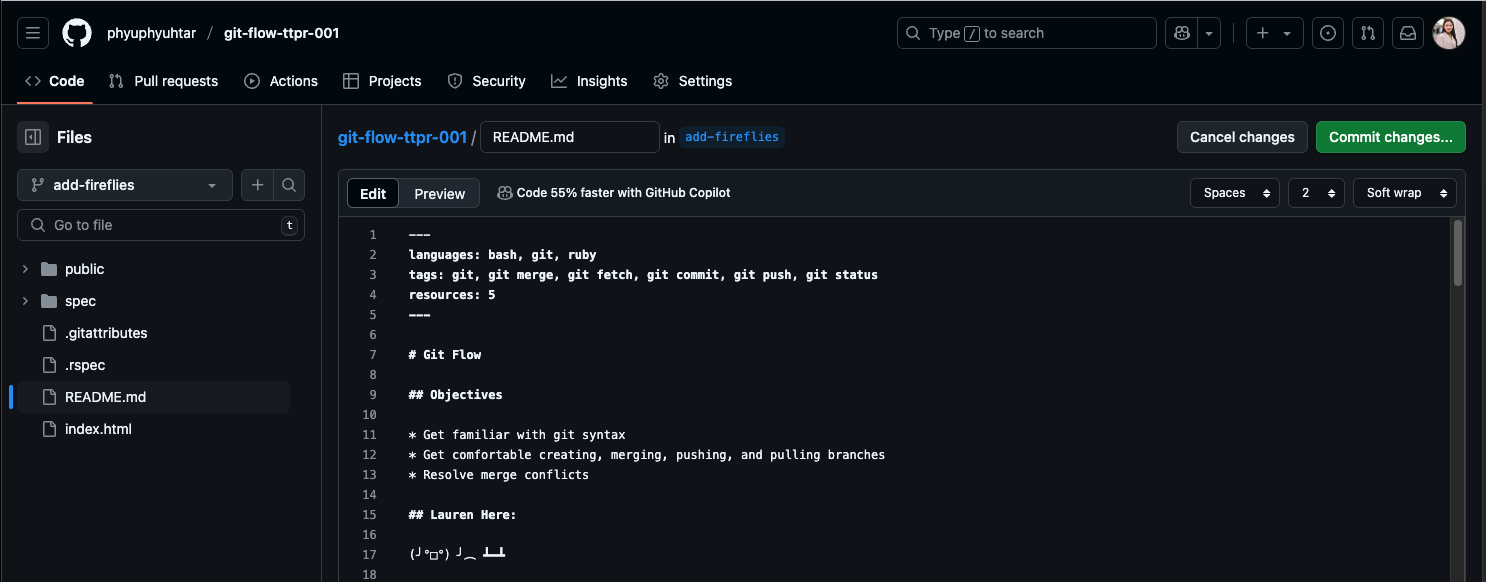
Now the readme should look like this:

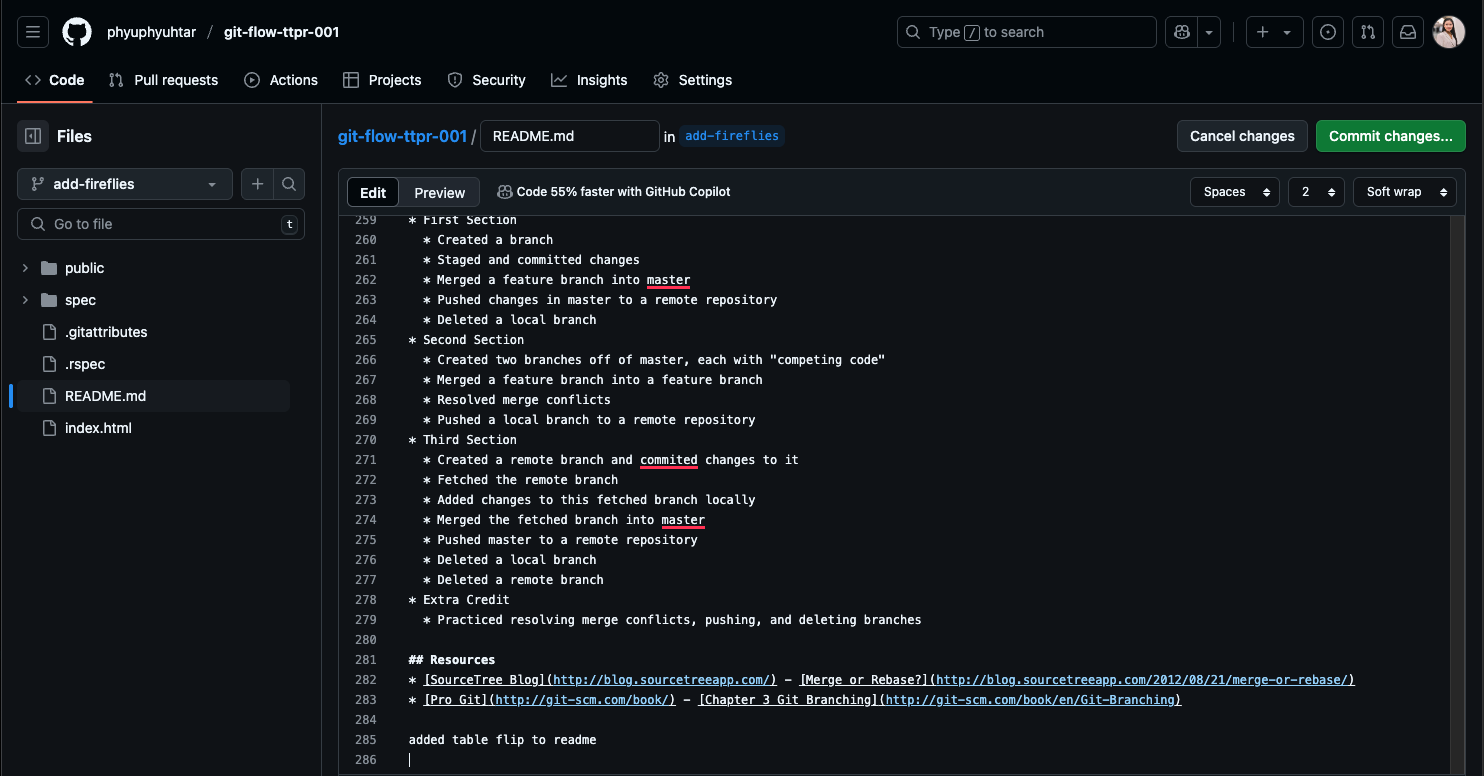
[](https://camo.githubusercontent.com/d74d4fa031c2aacf6eadf8de732a6981320970984bc0c45e7917e79f12b0d509/68747470733a2f2f73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f726561646d652d70686f746f732f6769742d666c6f772d7570646174652d6c617572656e2e706e67)

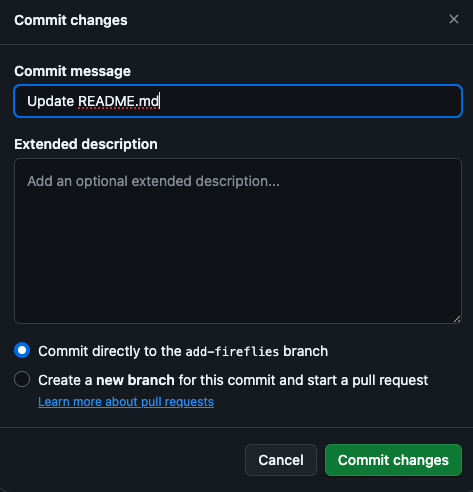
Scroll to the bottom, add a commit message like "added table flip to readme", and click commit changes.

[commit changes](https://camo.githubusercontent.com/fed39205488a3c8ef83f4593e04ca37a55cab2cfb616ced2ecf447407755915d/68747470733a2f2f73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f726561646d652d70686f746f732f6769742d666c6f772d7570646174652d636f6d6d69742d6368616e6765732e706e67)

Lauren's work here is done. You can go back to being you.







**You Time**

Now you need to get the changes that Lauren made.

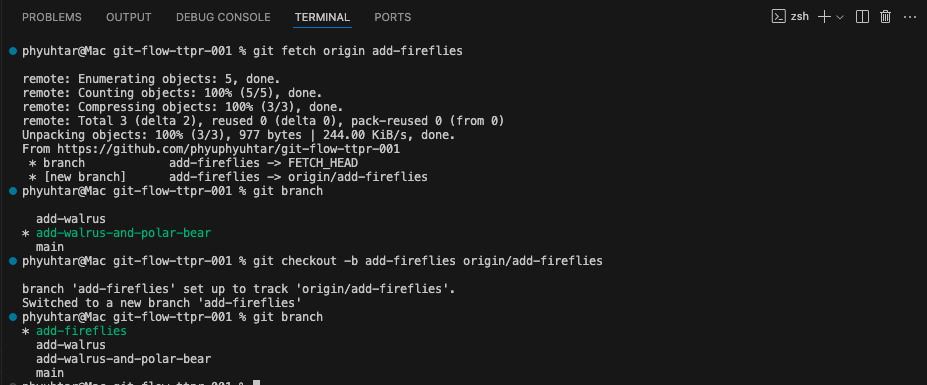
Update your remote-tracking branches by writing a fetch command: git fetch origin add-fireflies or git fetch --all

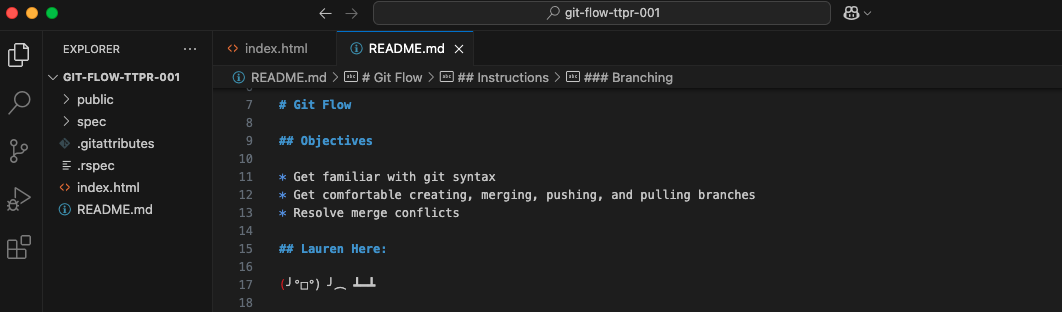
Type git branch. You probably still only see master and add-walrus-and-polar-bear. Think about why this is.

Checkout the branch Lauren created: git checkout add-fireflies

Now type git branch. You should see three branches: master, add-walrus-and-polar-bear, and add-fireflies (highlighted)

If you open the readme in Sublime from the add-fireflies branch, you should see Lauren's table flip addition.





Still in the add-fireflies branch, add the HTML code below to index.html, below the turtle.

<!-- begin fireflies picture and caption -->

<div class="center-container">

<div class="card">

<img src="public/img/fireflies.jpg" alt="fireflies in a forest">

</div>

<div class="card">

<div class="caption">

<h4>Searching for Love</h4>

<p>Spencer Black</p>

<p>Blue ghost fireflies in Brevard, North Carolina</p>

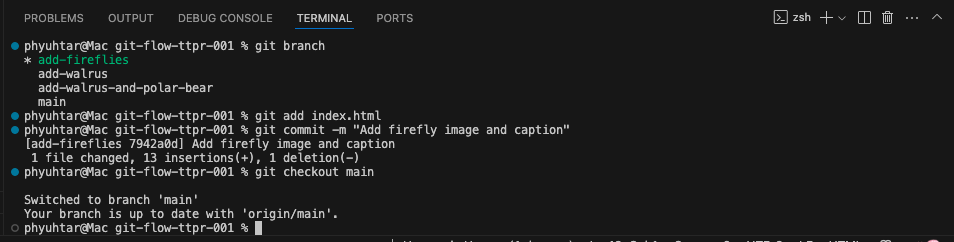
</div>

</div>

</div>

<!-- end fireflies picture and caption -->

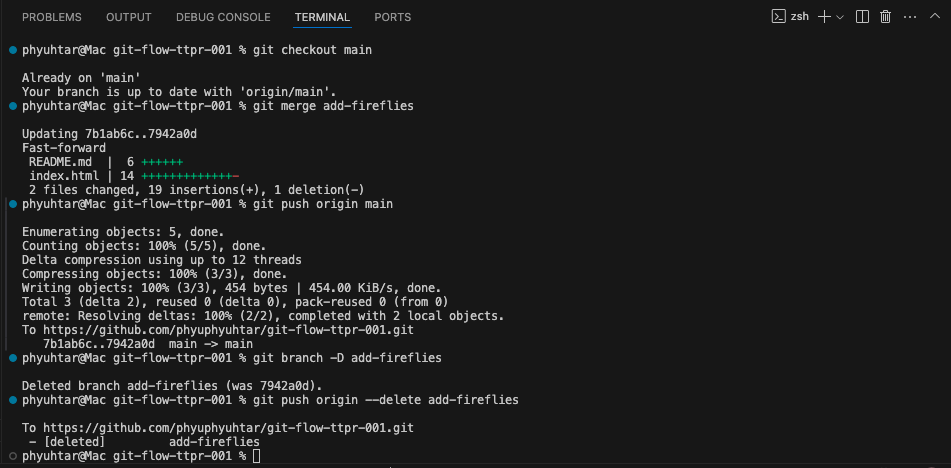
Add and commit these changes (git add index.html, git commit -m "add firefly img and caption") before checking out to master (git checkout master).



From master, merge add-butterflies (git merge add-fireflies). Make sure all three pictures are there (tree, turtle, fireflies) before pushing up to your remote repository (git push origin master).

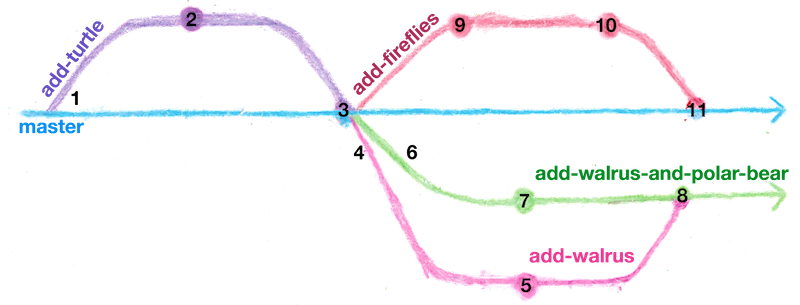
Since you merged into master and pushed the changes to a remote respository, delete the local branch (git branch -D add-fireflies).

Now it's time to delete the remote branch add-fireflies as it is now included in the master branch: git push origin :add-fireflies



**Visualization**

Just to review, this is the chronological order of this lab complete with a diagram of the branches:

[](https://camo.githubusercontent.com/617e0bcbfcafd53d8529020139e4b905f643a73e2704129fff3c3e2a530a37b8/68747470733a2f2f73332d75732d776573742d322e616d617a6f6e6177732e636f6d2f726561646d652d70686f746f732f6769742d666c6f772d7570646174652d6272616e636865732e706e67)

from master, create branch add-turtle

add turtle pic and caption to index.html

merge add-turtle into master, now master has tree and turtle pics

from master, create branch add-walrus

add walrus pic and caption

from master, create branch add-walrus-and-polar-bear

add polar bear pic and caption

merge add-walrus into add-walrus-and-polar-bear and fix merge conflicts - now branch has tree, turtle, polar bear, and walrus

"Lauren" makes branch add-fireflies off of master and adds an emoticon to the readme

add firefly pic and caption to add-firefly branch

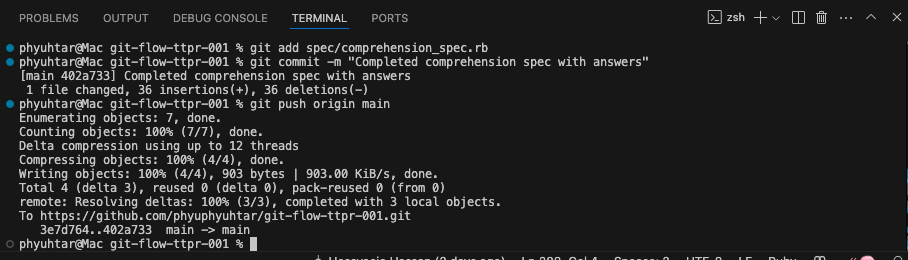
merge add-firefly into master

Without looking at the diagram above, try and draw your own visualization. Take a picture of your work and add that image to the public/img/ folder. Title your drawing "drawing.png".

main → add-turtle → add-walrus → conflict → add-fireflies,

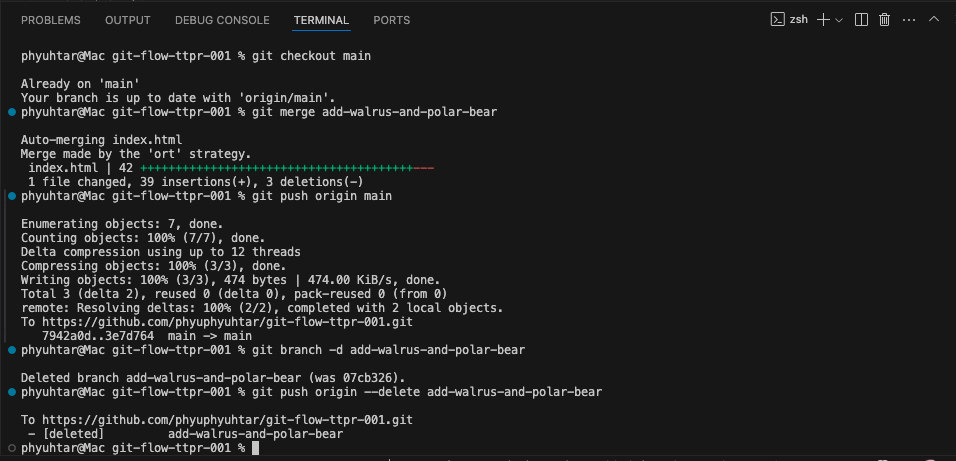
**Quiz Time**

From the master branch, replace the question marks in spec/comprehension\_spec.rb with the answers. Every answer should be a string. To check your answers, run rspec. Once every test is passing, submit a pull request using your master branch.

****

**Extra Credit**

Merge add-walrus-and-polar-bear into master. Fix the merge conflicts and push the changes before deleting the local and remote add-walrus-and-polar-bear branches.

****

**Conclusion**

Now you have:

First Section

Created a branch

Staged and committed changes

Merged a feature branch into master

Pushed changes in master to a remote repository

Deleted a local branch

Second Section

Created two branches off of master, each with "competing code"

Merged a feature branch into a feature branch

Resolved merge conflicts

Pushed a local branch to a remote repository

Third Section

Created a remote branch and commited changes to it

Fetched the remote branch

Added changes to this fetched branch locally

Merged the fetched branch into master

Pushed master to a remote repository

Deleted a local branch

Deleted a remote branch

Extra Credit

Practiced resolving merge conflicts, pushing, and deleting branches

**Resources**

[SourceTree Blog](http://blog.sourcetreeapp.com/) - [Merge or Rebase?](http://blog.sourcetreeapp.com/2012/08/21/merge-or-rebase/)

[Pro Git](http://git-scm.com/book/) - [Chapter 3 Git Branching](http://git-scm.com/book/en/Git-Branching)

[Pro Git](http://git-scm.com/book/) - [3.1 Git Branching - What a Branch Is](http://git-scm.com/book/en/Git-Branching-What-a-Branch-Is)

[Pro Git](http://git-scm.com/book/) - [3.2 Git Branching - Basic Branching and Merging](http://git-scm.com/book/en/Git-Branching-Basic-Branching-and-Merging)

[LearnGitBranching](http://pcottle.github.io/learnGitBranching/) - [Introduction Sequence: Branching in Git](http://pcottle.github.io/learnGitBranching/)