# Introduction

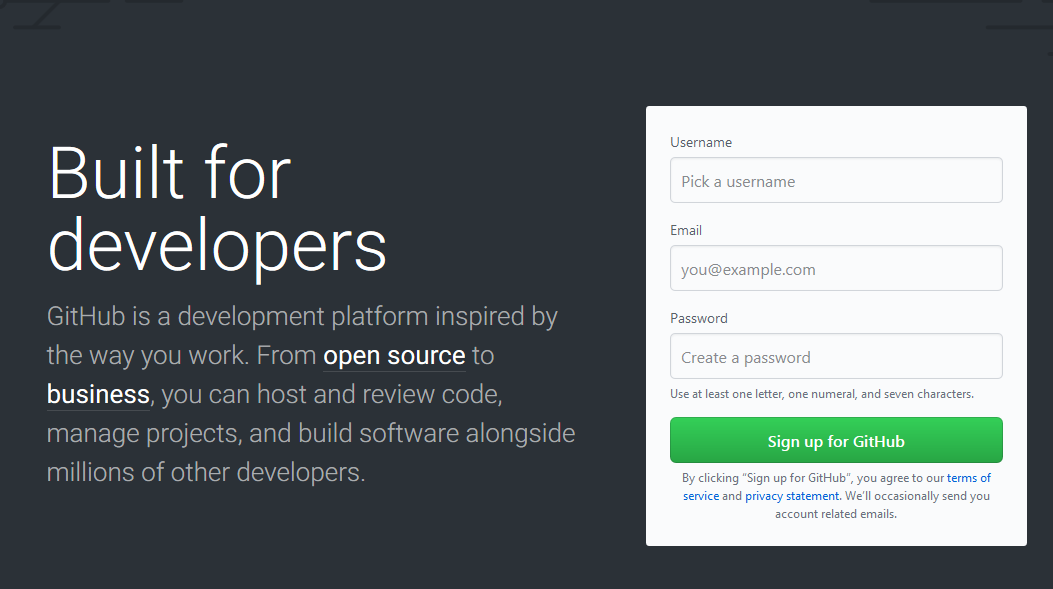
GitHub "is a web-based hosting service for version control using Git. It is mostly used for computer code." [https://en.wikipedia.org/wiki/GitHub]

Version control "is the management of changes to documents, computer programs, large web sites, and other collections of information." [https://en.wikipedia.org/wiki/Version\_control]

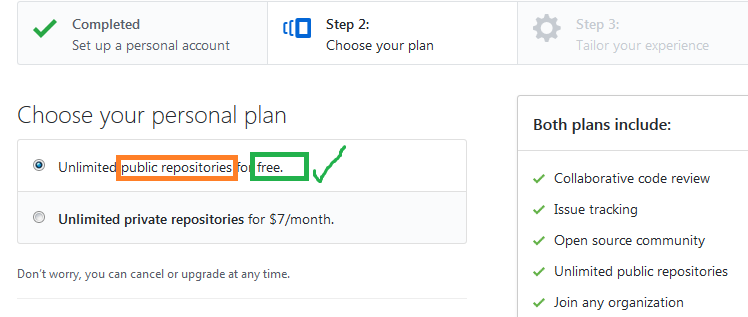
Git "is a version control system for tracking changes in computer files and coordinating work on those files among multiple people." [https://en.wikipedia.org/wiki/Git]

# Sign up for GitHub

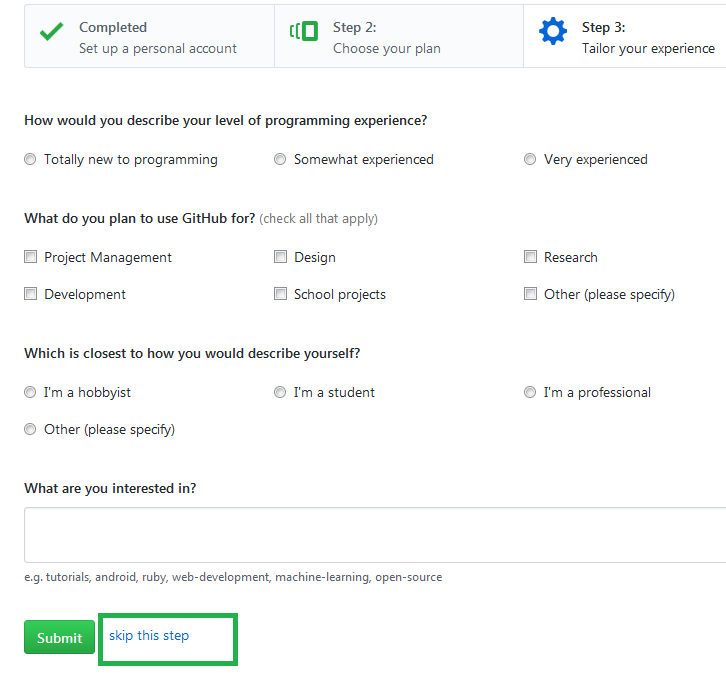
Go to https://github.com/



# Choose Your Plan



# Provide Training Data for GitHub Inc.'s Machine Learning?



# Resources

Excellent Tutorial for Git and GitHub...

https://blog.udacity.com/2015/06/a-beginners-git-github-tutorial.html

Excellent Introduction to GitHub...

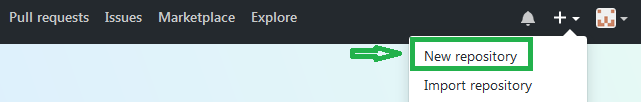
https://flaviocopes.com/github/

Excellent "Hello World" Guide from GitHub...

https://guides.github.com/activities/hello-world/

# Hello World

## Repository



A repository is an organization space.

## Branch

A branch is an instance of the stuff in a repository.

**master** is the default branch, typically considered the "master" branch.

New branches can be created (copied) from existing branches.

A common pattern is the following.

* **master** is used to perform definitive testing and to construct a software release.
* To work on a new feature, X, a new branch is created from **master**.
* The development team works on branch X.
* After feature X has been developed, tested, and approved, the work from branch X is merged into **master**.
* While X is being developed, other new features and/or bug fixes can have branches.

### Example: A Data Science Team Has Developed a Whale Identification Product

**master** representsa working product that was used for a Kaggle competition. The team was in the top 30%.

The team wants to improve the product to be in the top 20% for the next competition.

The team's assessment is that improvements are needed to (1) the image representation routine and (2) the automated hyperparameter tuning routine.

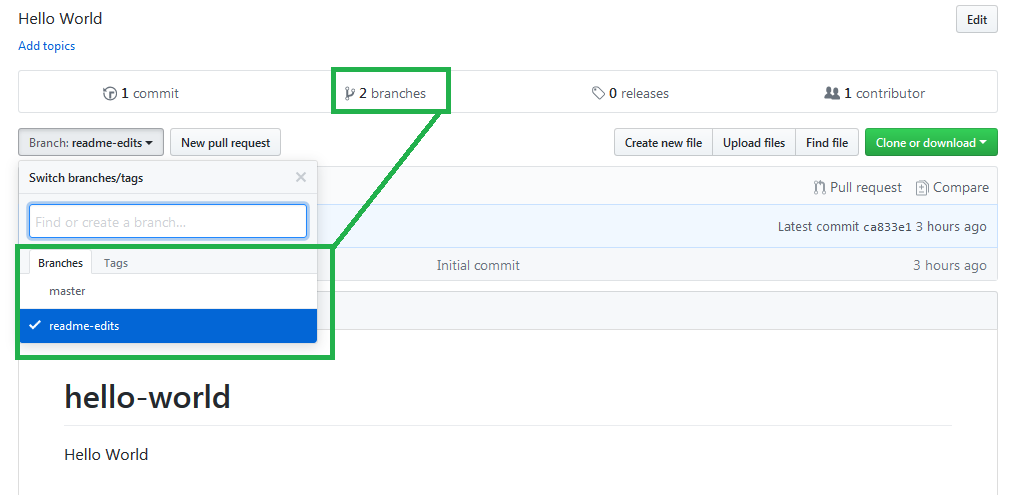
So the team creates two branches: (1) "Competition 2 Image Representation" and (2) "Competition 2 Hyperparameter Tuning."

The team works on those two branches.

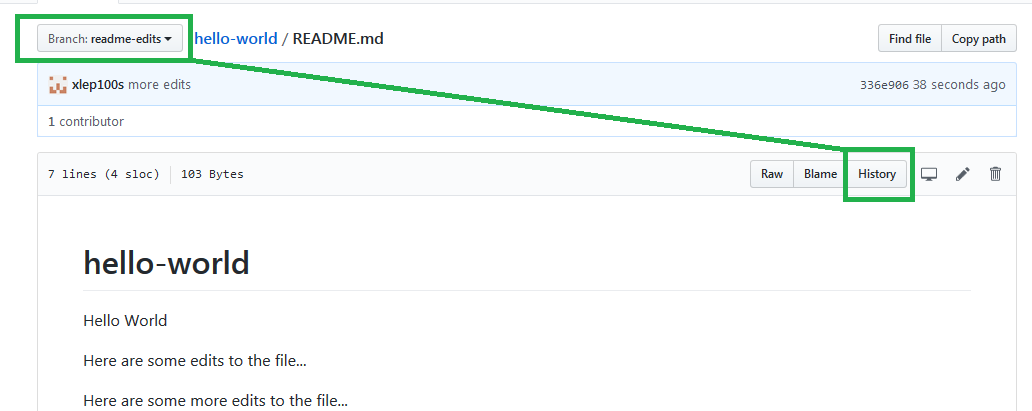
With time running out before having to submit for the competition, the team determines that the work done on the "Competition 2 Hyperparameter Tuning" branch improved the product but that the work on the "Competition 2 Image Representation" branch did not improve the product.

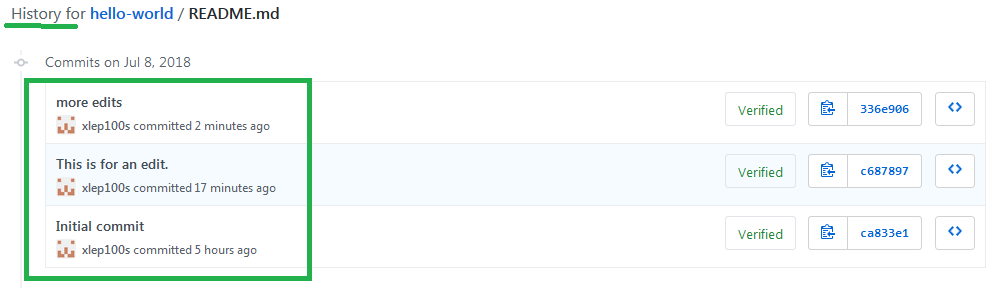
So "Competition 2 Hyperparameter Tuning" is merged into **master** and "Competition 2 Image Representation" is not.

## Create a New Branch

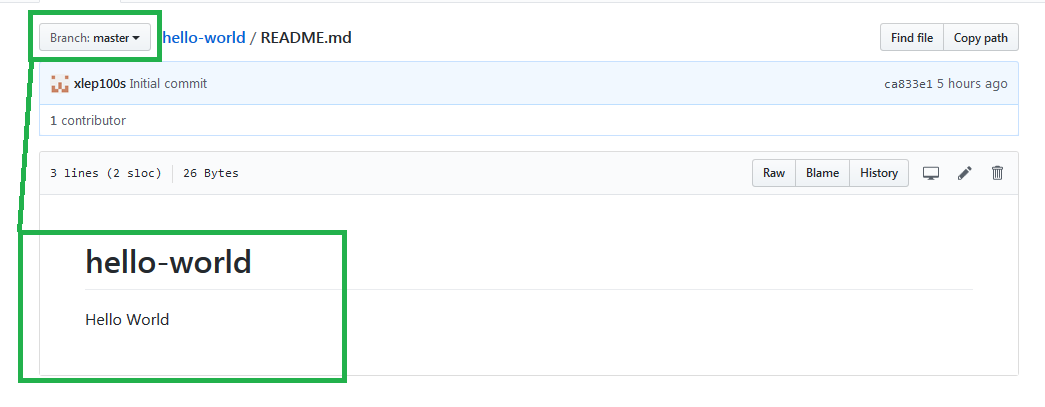


## Make and Commit Changes



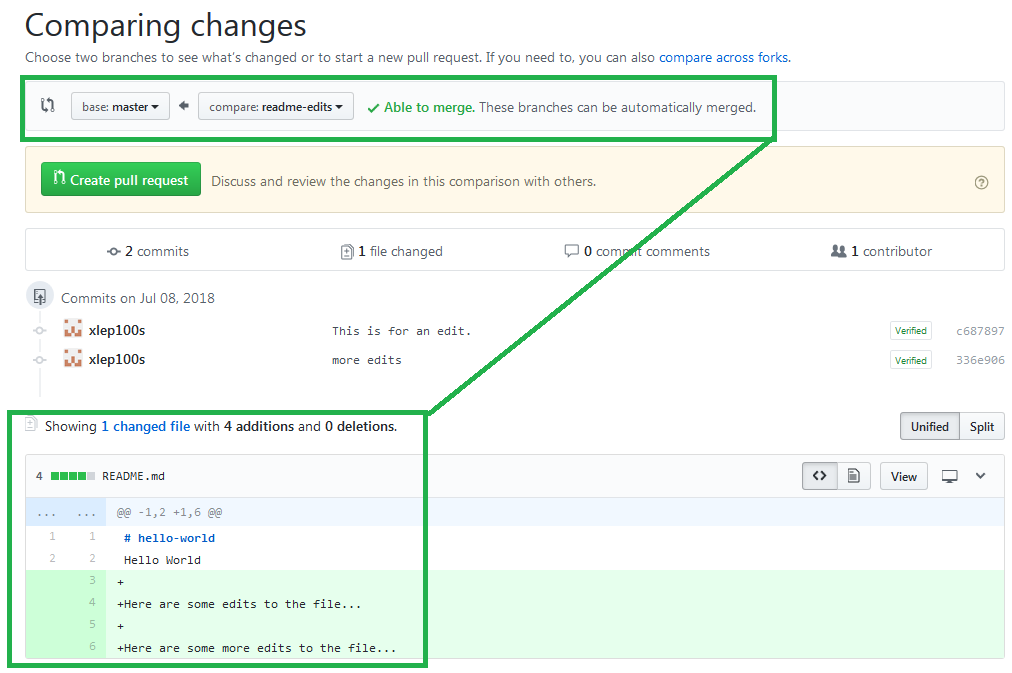


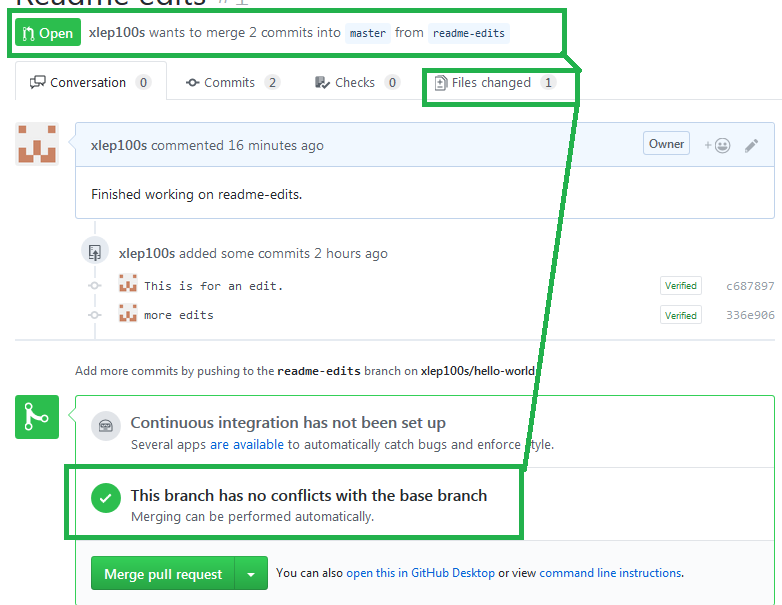
### master Does Not Have the Changes



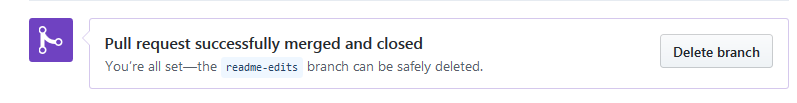
## Pull Request

Request that the changes on the **readme-edits** branch get pulled into the **master** branch.





## Merge



### master Has the Changes

