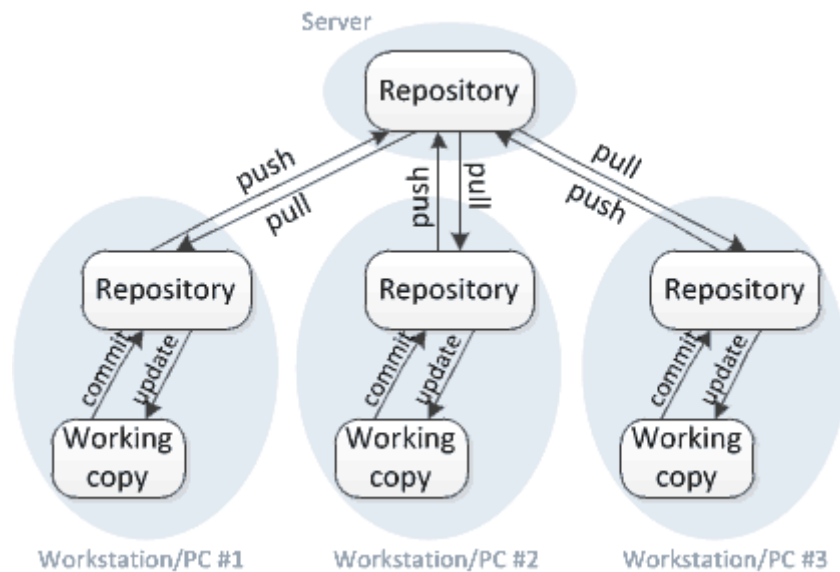


Version Control

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. GitHub is a form of Distributed Version Control Systems (DVCS), in a DVCS (such as Git, Mercurial, Bazaar or Darcs), clients don't just check out the latest snapshot of the files; rather, they fully mirror the repository, including its full history. Thus, if any server dies, and these systems were collaborating via that server, any of the client repositories can be copied back up to the server to restore it. Every clone is really a full backup of all the data.



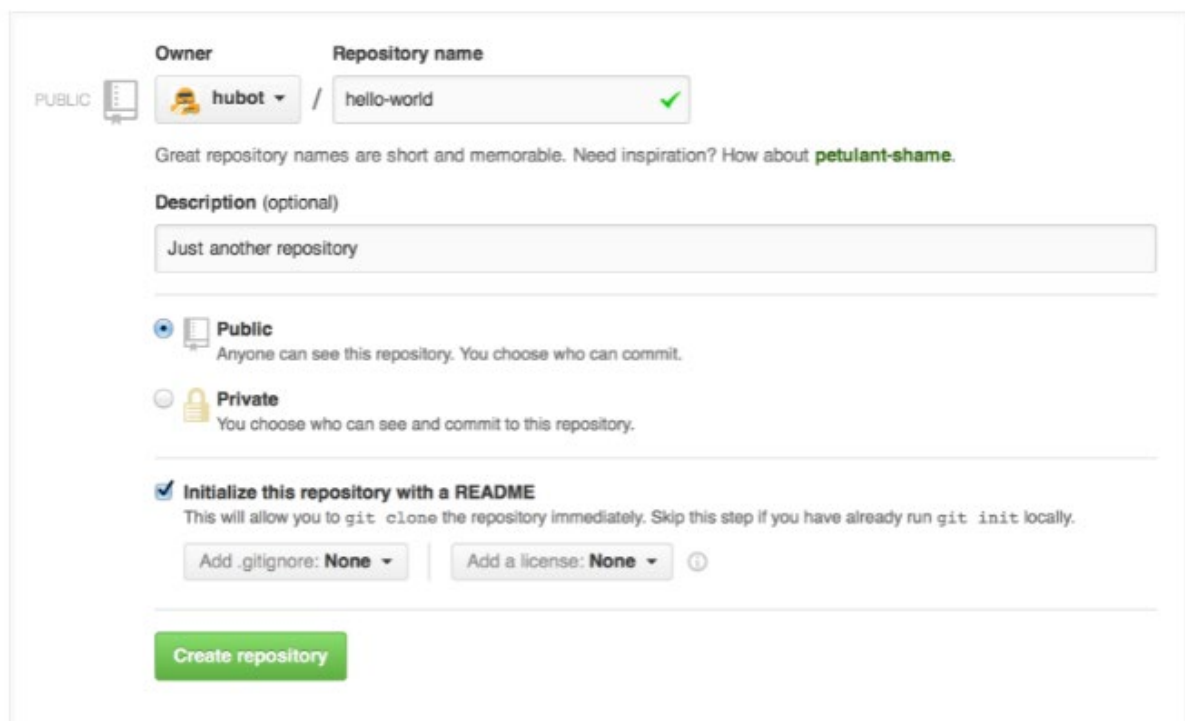
To complete this tutorial, you need a [GitHub](https://github.com) account and Internet access. You don't need to know how to code, use the command line, or install Git (the version control software GitHub is built on).

Step One – Create a Repository:

A repository is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets – anything your project needs. We recommend including a README, or a file with information about your project. GitHub makes it easy to add one at the same time you create your new repository. It also offers other common options such as a license file.

Your “*hello-world*” repository can be a place where you store ideas, resources, or even share and discuss things with others.

1. In the upper right corner, next to your avatar or identicon, click **+** and then select “New repository”.
2. Name your repository “hello-world”.
3. Write a short description.
4. Select Initialize this repository with a README.

A screenshot of the GitHub 'Create new repository' form. At the top, there's a 'PUBLIC' label and a 'Repository name' field containing 'hello-world' with a green checkmark. Below this is a 'Description (optional)' text area with the placeholder text 'Just another repository'. Further down, there are two radio button options: 'Public' (selected) and 'Private'. Below these is a checkbox labeled 'Initialize this repository with a README' which is checked. At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None'. A large green 'Create repository' button is at the very bottom.

5. Click “Create repository”.

Step Two – Create a Branch:

Branching is the way to work on different versions of a repository at one time.

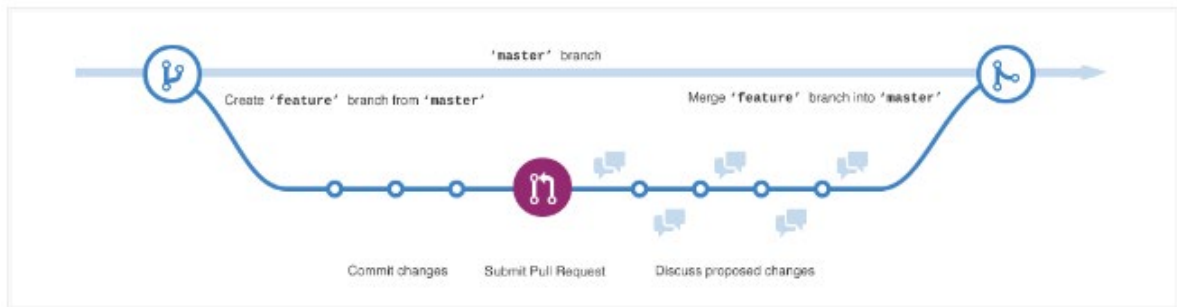
By default, your repository has one branch named “main” which is considered to be the definitive branch. We use branches to experiment and make edits before committing them to “main”.

When you create a branch off the “main” branch, you’re making a copy, or snapshot, of “main” as it was at that point in time. If someone else made changes to the “main” branch while you were working on your branch, you could pull in those updates.

This diagram shows:

- The main branch

- A new branch called feature (because we're doing 'feature work' on this branch)
- The journey that feature takes before it's merged into main



Have you ever saved different versions of a file? Something like:

- story.txt
- story-joe-edit.txt
- story-joe-edit-reviewed.txt

Branches accomplish similar goals in GitHub repositories.

To Create a New Branch:

- Go to your new repository "hello-world".
- Click the drop down at the top of the file list that says "branch: main".
- Type a branch name, "readme-edits", into the new branch text box.
- Select the Create branch box.

Just another repository — Edit

1 commit
1 branch

branch: master
hello-world / +

Initial commit

hubot authored just now

README.md
Initial

README.md

Now you have two branches, "main" and "readme-edits". They look exactly the same, but not for long! Next, we'll add our changes to the new branch.

Step Three – Make and Commit Changes:

Bravo! Now, you’re on the code view for your “readme-edits” branch, which is a copy of “main”. Let’s make some edits.

On GitHub, saved changes are called *commits*. Each commit has an associated *commit message*, which is a description explaining why a particular change was made. Commit messages capture the history of your changes, so other contributors can understand what you’ve done and why.

Make and Commit Changes:

1. Click the “README.md” file.
2. Click the pencil icon in the upper right corner of the file view to edit.
3. In the editor, write a bit about yourself.
4. Write a commit message that describes your changes.
5. Click **Commit changes** button.

The screenshot shows the GitHub interface for the 'hubot / hello-world' repository. The 'README.md' file is open in the editor. The commit message field is filled with 'Finish README' and 'And mention moon tacos'. The 'Commit changes' button is highlighted in green.

hubot / hello-world

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

hello-world / README.md or cancel

Edit file Preview changes Spaces 2 Soft wrap

```
1 # hello-world
2
3 Hi Humans!
4
5 Hubot here, I like Node.js and Coffeescript (that's what I'm made of!).
6 I've had tacos on the moon and find them far superior to Earth tacos.
7
```

Commit changes

Finish README

And mention moon tacos

☒ Commit directly to the `readme-edits` branch

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

These changes will be made to just the README file on your “readme-edits” branch, so now this branch contains content that’s different from “main”.

Step Four – Open a Pull Request

Nice edits! Now that you have changes in a branch off of “main”, you can open a *pull request*.

Pull Requests are the heart of collaboration on GitHub. When you open a *pull request*, you’re proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch. Pull requests show *diffs*, or differences, of the content from both branches. The changes, additions, and subtractions are shown in green and red.

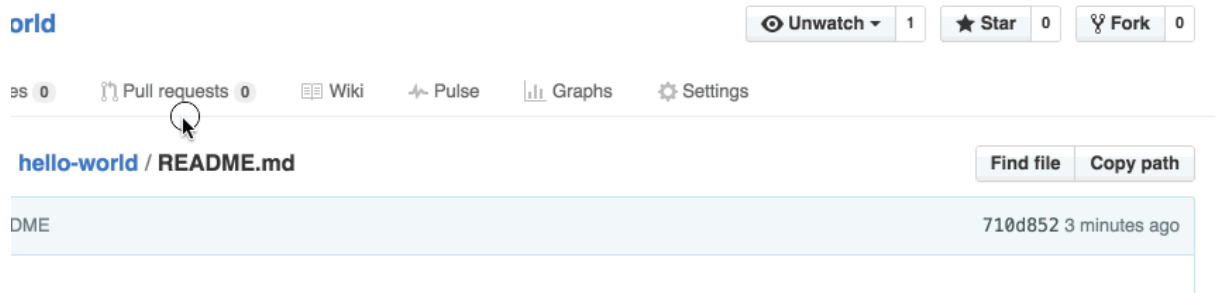
As soon as you make a commit, you can open a pull request and start a discussion, even before the code is finished.

By using GitHub's @mention system in your pull request message, you can ask for feedback from specific people or teams, whether they're down the hall or 10 time zones away.

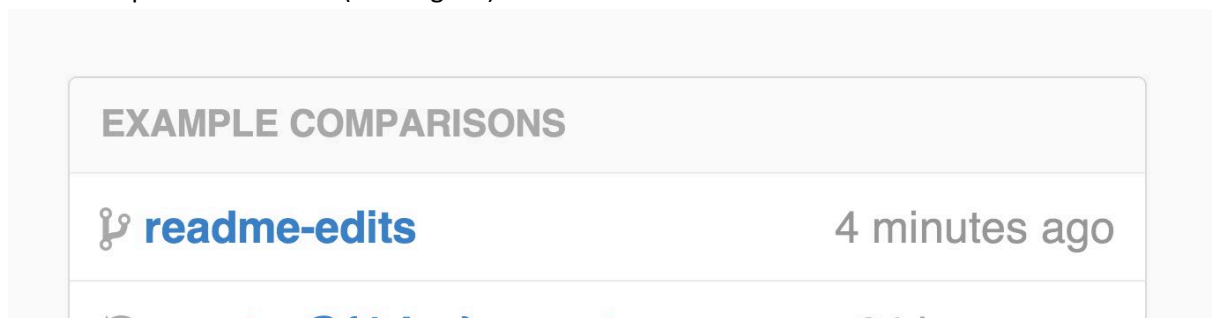
You can even open pull requests in your own repository and merge them yourself. It's a great way to learn the GitHub flow before working on larger projects.

Open a Pull Request for Changes to the README

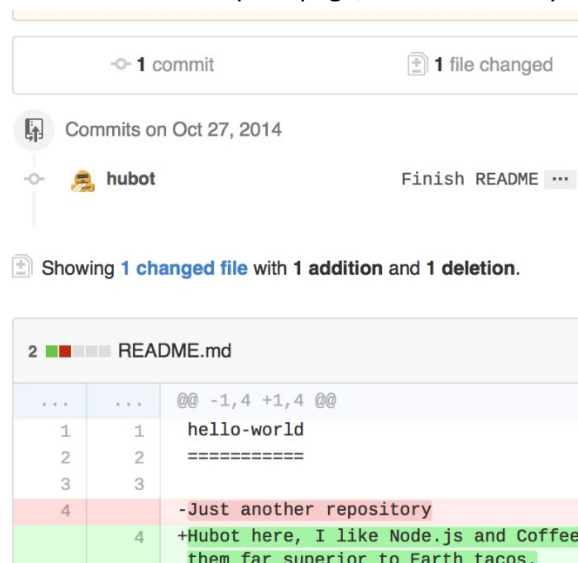
1. Click the "Pull Request" tab, then from the Pull Request page, click the green "New pull request" button.



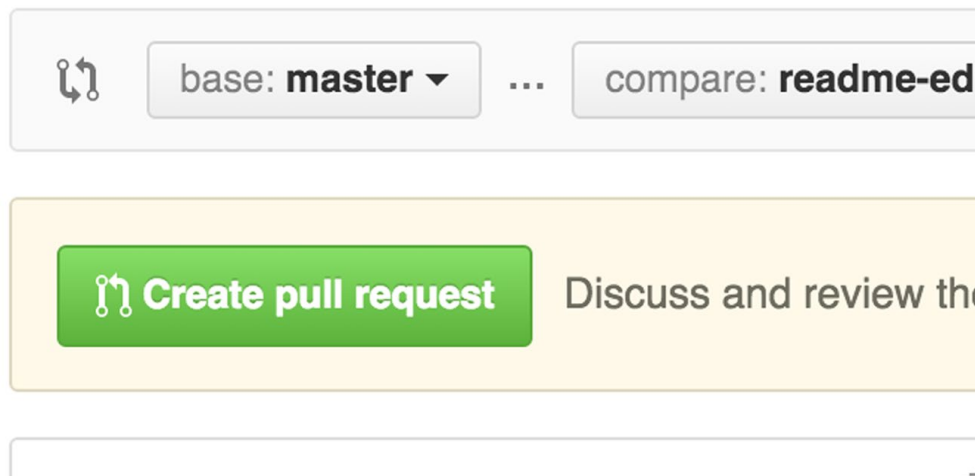
2. In the **Example Comparisons** box, select the branch you made, "readme-edits", to compare with "main" (the original).



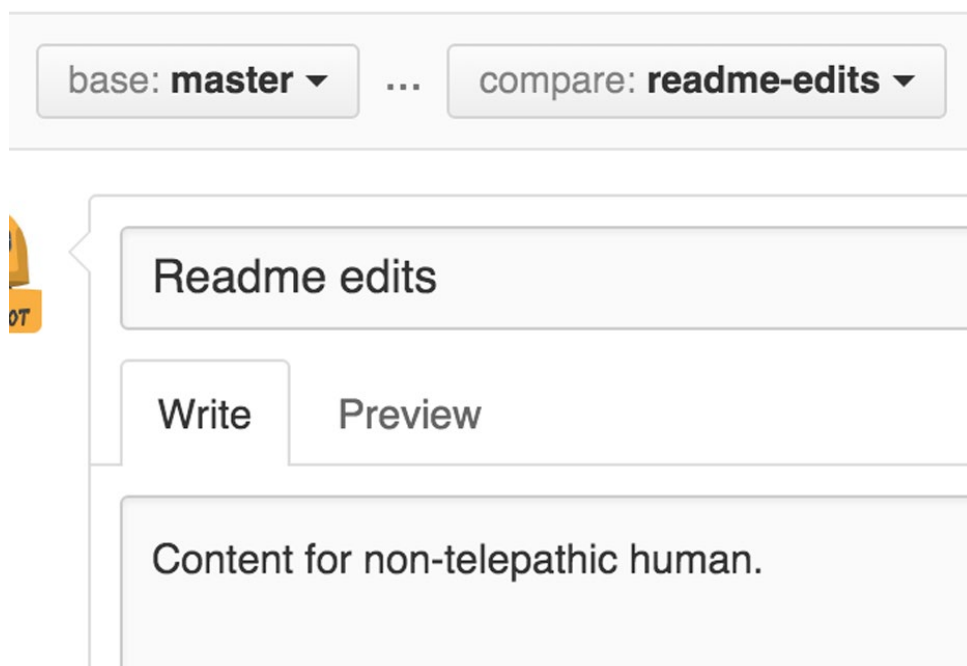
3. Look over your changes in the diffs on the Compare page, make sure they're what you want to submit.



4. When you're satisfied that these are the changes you want to submit, click the big green **Create Pull Request** button.



5. Give your pull request a title and write a brief description of your changes.





When you're done with your message, click **Create pull request!**

Step Five – Merge Your Pull Request


In this final step, it's time to bring your changes together – merging your readme-edits branch into the “main” branch.


1. Click the green **Merge pull request** button to merge the changes into main.
2. Click **Confirm merge**.
3. Go ahead and delete the branch, since its changes have been incorporated, with the **Delete branch** button in the purple box.



**This branch has no conflicts with the base branch**


Merging can be performed automatically.

 **Merge pull request** You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



Pull request successfully merged and closed

You're all set—the `readme-edits` branch can be safely deleted.

 **Delete branch**

Here's what you accomplished in this practical:

- Created an open-source repository
- Started and managed a new branch
- Changed a file and committed those changes to GitHub
- Opened and merged a Pull Request