How to get started with GitHub

Before Reading

What is GitHub?

GitHub is a website that allows you to visually look at and control your repositories as well as look at other repositories.

What is Git?

Git is a version-control software that allows a user to track the progress of anything. It is meant to be lightweight and easy to use so that everything is as fast and efficient as possible.

What is a repository?

A repository is a version-control space that allows you to perform Git actions. This is where you track progress/versions of whatever you are working on.

What is a branch?

A branch is a way to have multiple versions of your main repository that each have their own focus. Think of a repository as a tree with the trunk being the main branch. Merging the branches together allows for continuous progression.

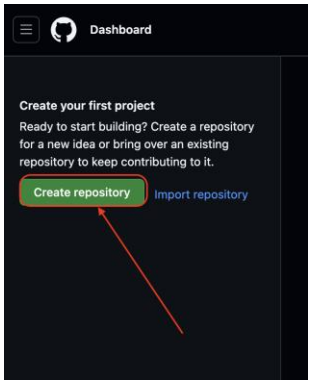
Why choose this guide?

This guide will show you how to get started, how to create a repository, and basic actions that you'll need to know. This guide is aimed at new computer science students at the university level.

Getting Started

How to create a repository:

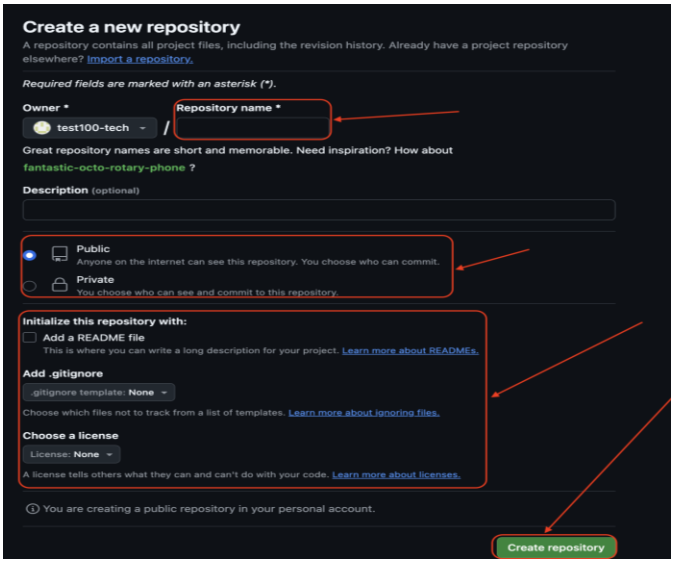
1. First, open the website in your web browser of choice.
2. After opening the website, create an account by signing up.
3. Sign back in using the information that you entered and now we're ready to get started.
4. We are now going to begin by creating a repository. On the left-hand side, click "Create repository".



5. Click "Create repository" and begin by entering a name for this repository. It doesn't matter what the name is for now, but in the future name them so that they match the contents.

6. Then, you can select either “Public” or “Private” for the type of repository. Currently it doesn’t matter but, in the future, you can select either based on what the purpose of the repository is. “Public” as it implies, makes it so that anyone can view it, but “Private” makes it so only you can view it.

7. You can initialize your repository with some extra content such as a README, a .gitignore, and a license. They are not always needed but are useful.
    - README - A file that explains the repository’s purpose
    - .gitignore - A template that ignores certain files from being tracked
    - Liscense - Protects your work from being stolen
8. For now, don’t include them just to keep things simple.
9. After configuring the repository to your liking click “Create repository”.

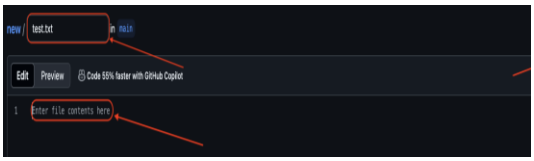


- Note: At this point it is recommended that you are at least somewhat familiar with Git and the terminal but if not, you can still proceed.

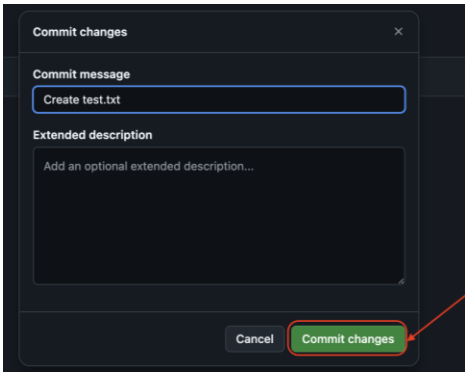
How to add a file and make changes:

1. Begin by clicking “create a new file”.

2. Enter your desired file name and type. This can differ based on what you are doing, but for now a .txt file is sufficient. Now, simply write anything you want. You can write an essay for school or code a new program. For now, keep it short so that you can learn the basics.



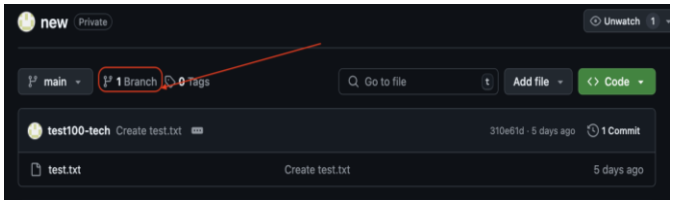
3. Now we can commit, or save, these changes that you made. Click “Commit changes…”. A window will pop up allowing you to enter a commit message. Enter anything you’d like, but make sure it accurately describes the changes you made. The heading should be short, and the extended description is where you be more in depth.
4. After entering your desired commit message, click “Commit changes”. Now you have created your first file and committed some new changes. This is the basis of Git and GitHub but there are a few other actions that we will cover.



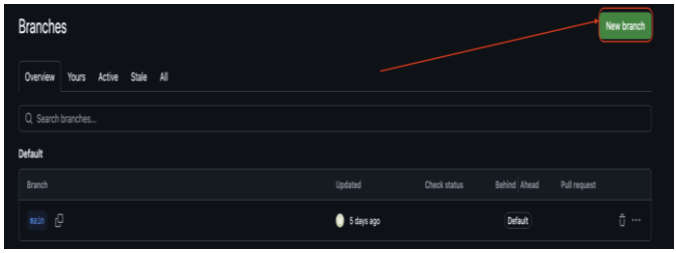
- **Note**: These changes are final, so make sure you are sure that everything looks good before you commit.

How to make a branch:

1. We will make a new branch in our repository. Until now, we have been working on the ‘main’ branch. Making new branches is advantageous because it allows you to focus on different parts of your overall project (e.g. bugs or new features).
2. Select the “branch” icon with the words “1 Branch” next to it.

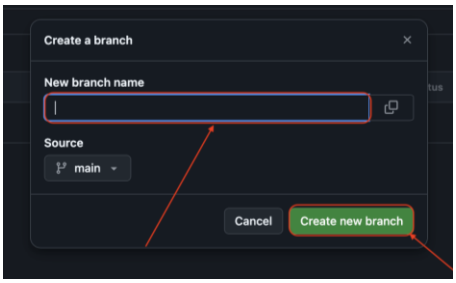


3. This is going to bring you to the page where you can manage all of the branches in your repository. Now, select the “New Branch” button so we can now make a new branch.



4. You want to enter a name for the branch that describes its purpose. For example, “Back-end Bug” or something so that you and others can easily know what this branch is focused on.

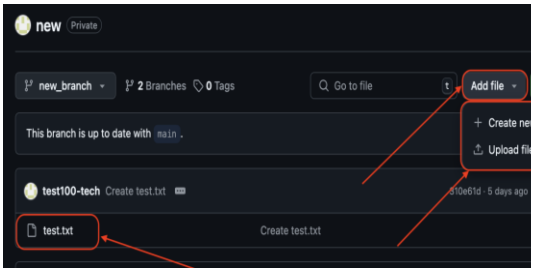
5. Then, select “Create new branch”.



6. Press the “<> Code” button to go back to the main page of your directory. Press the drop down where it says “main” so that we can now switch to the newly created branch. In Git, you can only work in one branch at a time. You have to checkout (switch) to each branch manually.

- **Note**: Make sure you are on the correct branch that you want to be before making any changes.

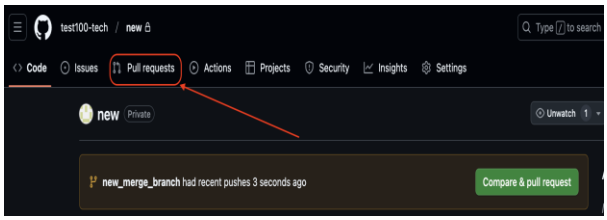
7. Now in the drop down, select the newly created branch so that we can switch to it.



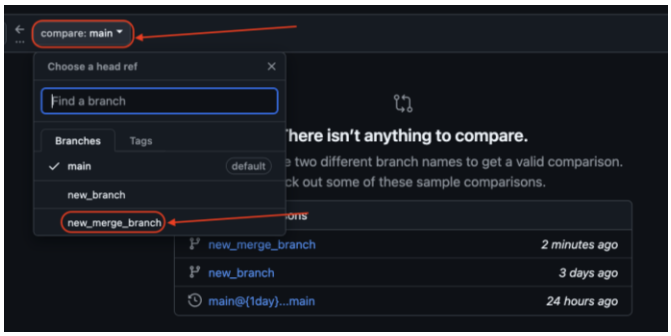
8. In the new branch, we can follow the same previous steps of adding/creating a new file or we can edit an existing file. The options are limitless and completely up to you.

How to merge a branch:

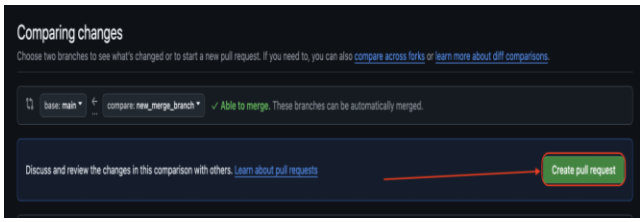
1. We are now ready to merge the new changes into our main branch, thus making them official.
2. Begin by selecting “Pull requests” from the navigation bar.



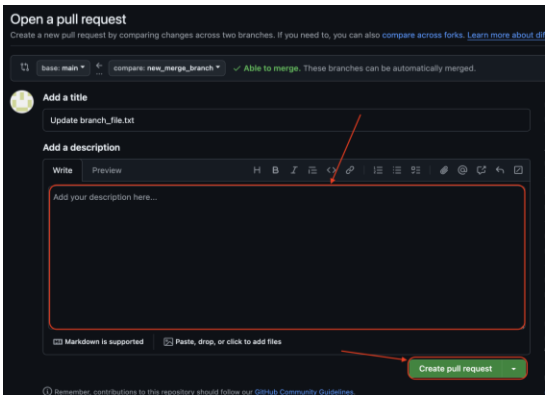
3. Then click the drop-down menu and select the branch that you want to merge into main.



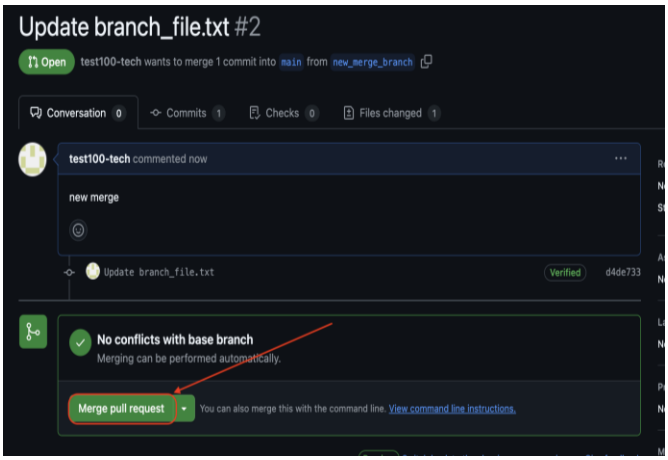
4. Select “Create new pull request”. This is making a formal request to see if it’s okay to merge your selected branch into main. In the real world, there can be conflicts when doing a merge.



5. You are going to write a description before selecting “Create pull request”.



6. Select “Merge pull request” to finalize the merge.



Conclusion

After following these instructions, you will successfully know how to use GitHub and basic Git action. These instructions can be applied to your learning in your courses which will then translate to the job field.