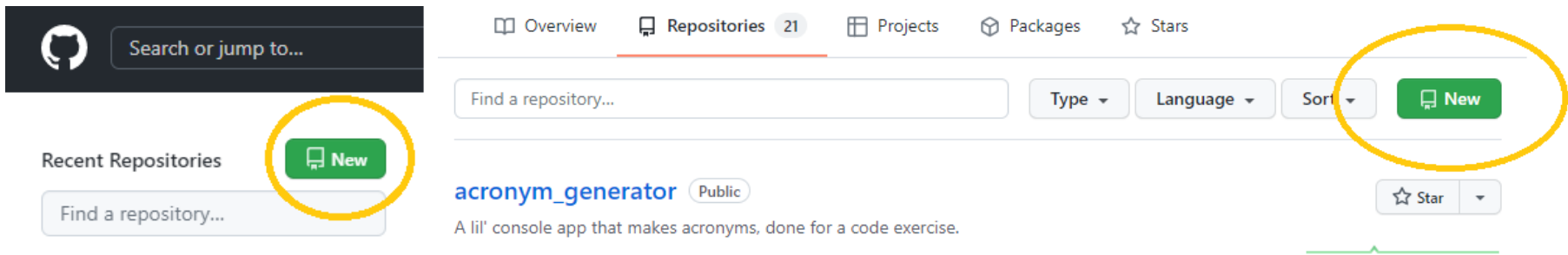


How to Push Your Tech Elevator Repositories to Your Personal GitHub Account

- Follow these steps for each repository you would like to move to GitHub.
- *Note: Once you follow these steps, your code repository will be disconnected from Tech Elevator.*
- First, you'll need a [GitHub](#) account! Create an account on GitHub (github.com), log in, and then follow these steps to move your repository.

Step One: Create a repository on GitHub.

You can find the “new” repository button on the left-hand side of the page when you log into your account. Or, you can click on your avatar in the upper right corner of the site, and select “your repositories” from the dropdown menu. Once you’re on your repository page, the green “new” button will be to the right of the search bar at the top:



Click the “new” button, and then fill out the form.

- The **repository name** should match the name of your repository in C:/Users/Student/git.
- Choose **private**, unless you want to make the content of the repository available to the public (you can change this setting later).
- You do not need to change any other inputs on the form.
- When finished, click the “create repository” button at the bottom of the form.

Owner *

tbrenneison

Repository name *

/ module-1-capstone-main

Great repository names are short and memorable. Need inspiration? How about [fantastic-octo-broccoli](#)?

Description (optional)

☐ Public

Anyone on the internet can see this repository. You choose who can commit.

☒ Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☒ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

gitignore template: None

☒ Choose a license

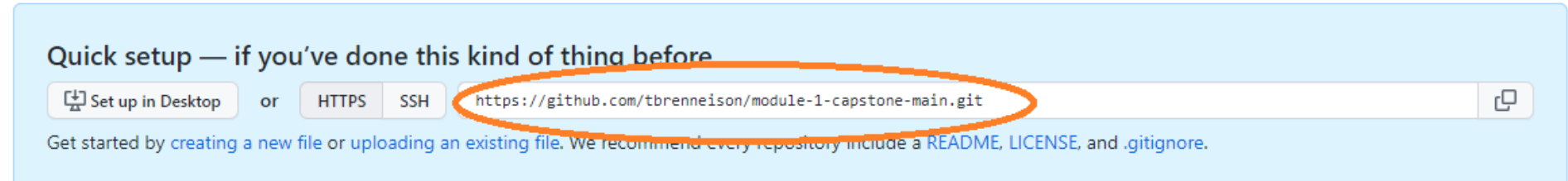
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None

You are creating a private repository in your personal account.

Create repository

Copy the URL of your new GitHub repository.



Step Two: Push your local repository to the new GitHub repository.

Open Git Bash, and go to the repository that you want to move to GitHub.

```
MINGW64:/c/Users/Student/git/module-1-capstone-main

Tori@V-P88F001 MINGW64 ~/git/module-1-capstone-main (main)
$ |
```

Rename the origin repository to ‘gitlab’: `git remote rename origin gitlab`

Using the copied URL from GitHub, add the GitHub remote repository as the new origin repo: `git remote add origin https://github.com/tbrenneison/module-1-capstone-main.git`

Push your repository: `git push origin main`

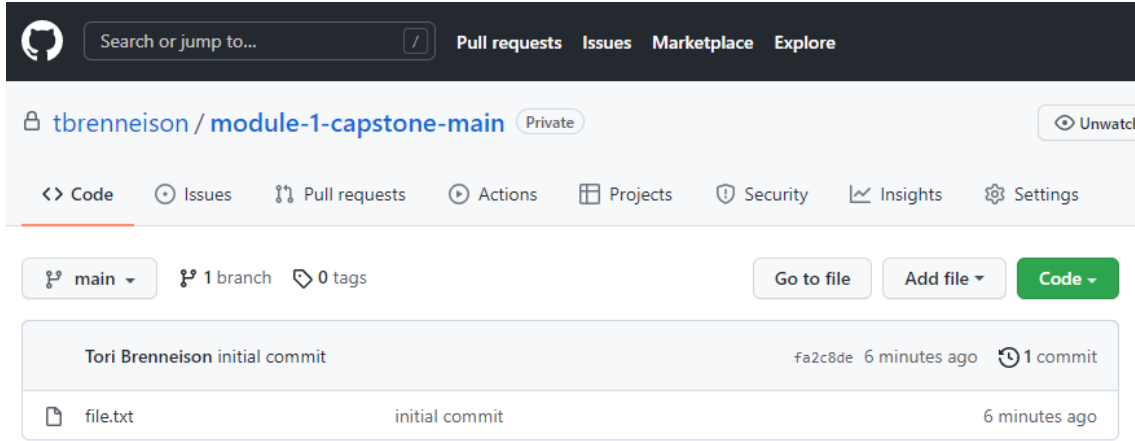
```
Tori@V-P88F001 MINGW64 ~/git/module-1-capstone-main (main)
$ git remote rename origin gitlab

Tori@V-P88F001 MINGW64 ~/git/module-1-capstone-main (main)
$ git remote add origin https://github.com/tbrenneison/module-1-capstone-main.git

Tori@V-P88F001 MINGW64 ~/git/module-1-capstone-main (main)
$ git push origin main
```

Note: You may be asked to authenticate your account in some way before pushing.

Go to the URL of your new repository on GitHub and make sure it looks correct.



Repeat these steps for each repository (capstones, etc.) from your Tech Elevator GitLab account.

You will need to log into GitHub from your personal computer, and clone the repositories to your personal computer in order to access them locally.

Authenticating your GitHub account in order to push a repository:

Git Credential Manager:

You may be asked to use the ‘Git Credential Manager’ in your web browser to authorize access to your account when you push. Click the green button to enter your GitHub password if necessary, and the push will continue.

GitHub Credentials in Git Bash:

You may be asked to put your username and password for GitHub into Git Bash. Put in your username as usual, and hit enter.

Once you are prompted for your password, you will not be able to see your cursor or what you have typed into the shell. The input is still there, but it is hidden as a security measure. If you feel as though you have made a mistake typing in your password, make sure to backspace enough to completely clear out the password field (that you can’t see) and try your entire password again. Hit the enter key when you are done.

If you are prompted to use a Personal Access Token instead of your GitHub password, follow the instructions below to set up a Personal Access Token. The access token is used just like the account password in the bash shell.

- Note: Ctrl + V to paste does not work in the bash shell. To paste your access token into the shell, right-click in the shell and select “paste” from the dropdown menu.

Personal Access Tokens:

GitHub may ask you to use a Personal Access Token instead of a password to authenticate your account. To set up a Personal Access Token, go to GitHub in your web browser, click your user avatar in the upper right corner of the site, and select “settings” from the dropdown menu. Click on ‘developer settings’ at the very bottom of the left side menu, and then select “personal access tokens” from the resulting page’s side menu.

- Click on “generate new token”, and fill out the form.
- Put a descriptive note in the ‘note’ input.
- You can generate a token that does not expire, or set an expiration date for your token.
- Check the ‘repo’ checkbox to use your token for working with repositories.

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

pushing to github

What’s this token for?

Expiration *

No expiration The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/>	repo	Full control of private repositories
<input checked="" type="checkbox"/>	repo:status	Access commit status
<input checked="" type="checkbox"/>	repo_deployment	Access deployment status
<input checked="" type="checkbox"/>	public_repo	Access public repositories
<input checked="" type="checkbox"/>	repo:invite	Access repository invitations
<input checked="" type="checkbox"/>	security_events	Read and write security events

Click the “generate new token” checkbox at the bottom of the form.

Make sure that you **copy the generated token** on the resulting page. You will not be able to copy it again, and will have to generate a new token if you lose it.

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won’t be able to see it again!

✓ ghp_ODC		Delete
-----------	--	--------