Github

Let's go through the steps of creating a Git repository, adding files, staging, committing, and pushing to GitHub.

**1. Create a Local Git Repository:**

First, you need a local directory for your project. Let's call it my-git-project.

Bash

mkdir my-git-project

cd my-git-project

Now, initialize a Git repository in this directory:

Bash

git init

This creates a hidden .git folder, which is the heart of your Git repository.

**2. Create the Files:**

Create the three files: Code.txt, Log.txt, and Output.txt. You can use any text editor or the command line:

Bash

touch Code.txt Log.txt Output.txt

Add some content to each file. For example:

* **Code.txt:**

<!-- end list -->

This is some code.

* **Log.txt:**

<!-- end list -->

This is a log file.

* **Output.txt:**

<!-- end list -->

This is the output.

**3. Stage the Code.txt and Output.txt Files:**

Staging means preparing the files for commit. Use the git add command:

Bash

git add Code.txt Output.txt

You can verify which files are staged using:

Bash

git status

You should see Code.txt and Output.txt listed under "Changes to be committed". Log.txt will be under "Untracked files".

**4. Commit the Staged Files:**

Now, commit the staged files with a descriptive message:

Bash

git commit -m "Added code and output files"

The -m flag is used to provide a commit message.

**5. Create a Repository on GitHub:**

1. Go to [https://github.com/](https://www.google.com/url?sa=E&source=gmail&q=https://github.com/) and log in (or create an account if you don't have one).
2. Click the "+" button in the top right corner and select "New repository".
3. Give your repository a name (e.g., my-git-project). You can add a description if you want.
4. Choose whether you want the repository to be public or private.
5. Click "Create repository".

GitHub will provide you with instructions on how to push an existing repository. You'll see something like this (replace your\_username and my-git-project with your details):

Bash

git remote add origin https://github.com/your\_username/my-git-project.git

git branch -M main

git push -u origin main

**6. Push to GitHub:**

Use the commands provided by GitHub (as shown above):

* git remote add origin https://github.com/your\_username/my-git-project.git: This adds a remote named "origin" that points to your GitHub repository.
* git branch -M main: Renames the local branch to main. (Older versions of Git might use master as the default branch name.)
* git push -u origin main: This pushes your local commits to the main branch on the "origin" remote (your GitHub repository). The -u flag sets up tracking, so you can just use git push in the future.

**Complete Sequence of Commands:**

Bash

mkdir my-git-project

cd my-git-project

git init

touch Code.txt Log.txt Output.txt

# Add content to the files

git add Code.txt Output.txt

git commit -m "Added code and output files"

# On GitHub, create a new repository (e.g., my-git-project)

git remote add origin https://github.com/your\_username/my-git-project.git

git branch -M main

git push -u origin main

After these steps, your Code.txt and Output.txt files (and the commit history) will be on your GitHub repository. The Log.txt file will remain in your local directory but will not be tracked by Git because you did not stage it. If you want to add it later, you can use git add Log.txt, git commit -m "Added log file", and git push.