

# Github 101

By Irene :) 2/28/2023

GitHub can be a valuable tool for anyone learning data analytics. Here are a few reasons why:

1. **Collaboration:** GitHub allows you to collaborate with others on projects. This can be particularly useful if you're working on a data analytics project with others. By using GitHub, you can share your work with others and collaborate on code changes in real-time.
2. **Version Control:** GitHub allows you to track changes to your code over time, making it easier to see how your analysis has evolved. This can be particularly useful if you're working on a long-term data analytics project, where you may want to go back and review earlier versions of your work.
3. **Showcasing Your Work:** GitHub provides a platform for showcasing your data analytics work to potential employers or clients. By creating a public repository and adding your analysis projects to it, you can build a portfolio that demonstrates your skills and experience.

If you're interested in showcasing your data analytics work on GitHub, here are a few tips:

1. **Create a Public Repository:** To showcase your work, you'll need to create a public repository on GitHub. This will allow others to view your code and analysis.
2. **Add README Files:** For each project in your repository, be sure to include a README file that provides an overview of the project and its purpose. This will help others understand your work and context.
3. **Use Jupyter Notebooks:** Jupyter notebooks are a great way to present your data analysis work. By adding Jupyter notebooks to your repository, you can showcase your data analysis in an interactive way.
4. **Include Visualizations:** Visualizations are a key component of data analytics. Be sure to include visualizations of your data in your analysis projects.
5. **Keep Your Repository Up-to-Date:** As you continue to work on new data analytics projects, be sure to update your repository with new code and analysis. This will keep your portfolio fresh and demonstrate that you're actively engaged in data analytics work.

## Learning Version Control w/ GitHub

### Step 1: Create a GitHub account

To use GitHub, you need to create an account first. Go to the GitHub website (<https://github.com/>) and sign up for an account. Follow the prompts to create your account.



## Step 2: Install Git

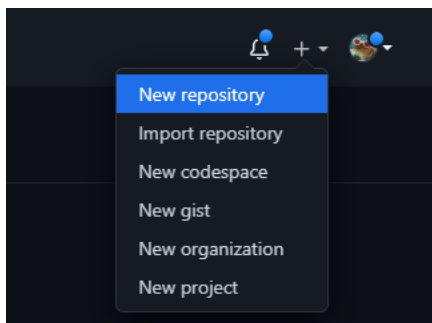
Git is a version control system that allows you to manage your code changes. GitHub is built on top of Git, so you need to install Git on your computer before you can use GitHub. You can download Git from the official website (<https://git-scm.com/downloads>).



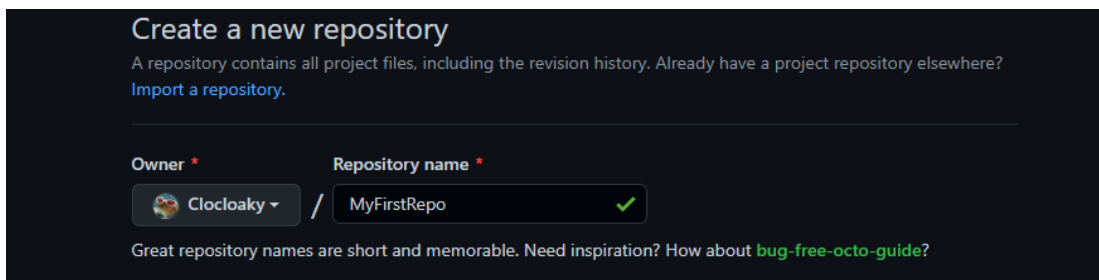
## Step 3: Create a new repository

After you've signed up for a GitHub account and installed Git, you can create a new repository. To create a new repository, follow these steps:

1. Click the "+" icon in the top-right corner of the GitHub website.



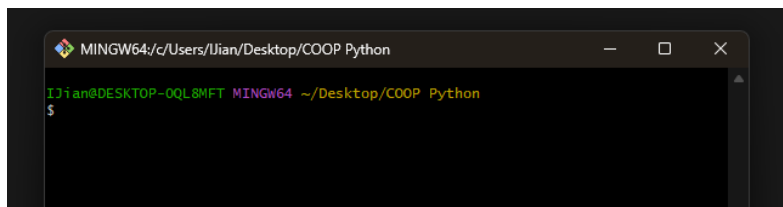
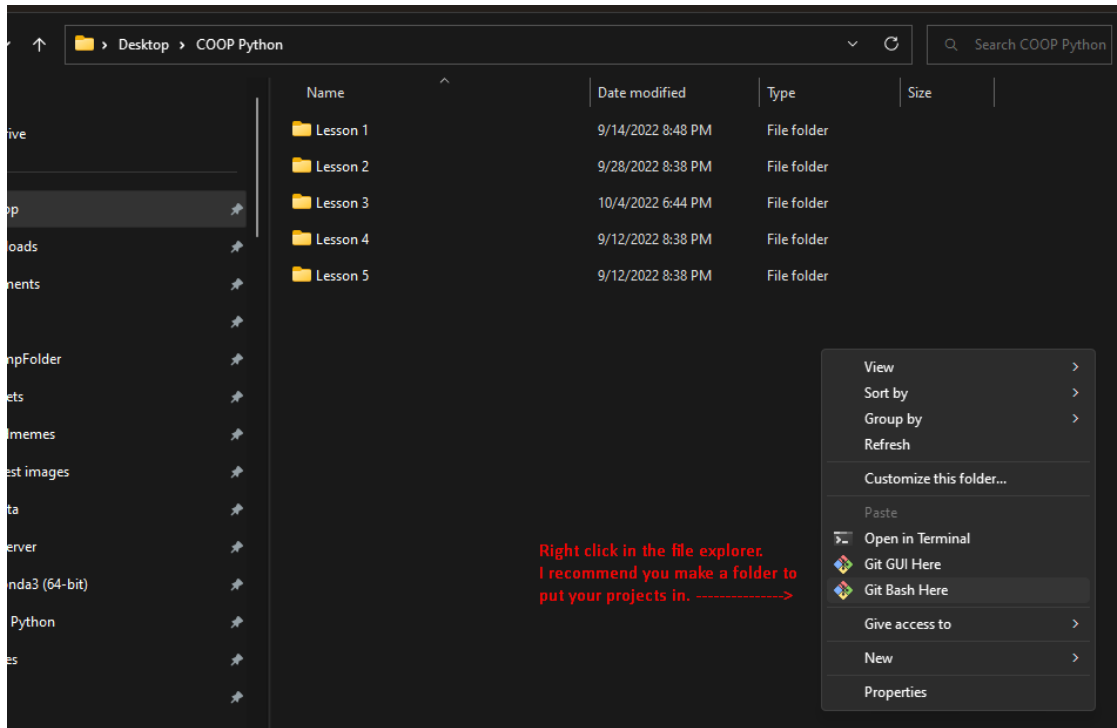
2. Select "New repository" from the dropdown menu.
3. Choose a name for your repository and select whether it should be public or private.
4. Choose whether to **include a README file (Checkmark this!)**, a .gitignore file, and a license.
5. Click "Create repository."



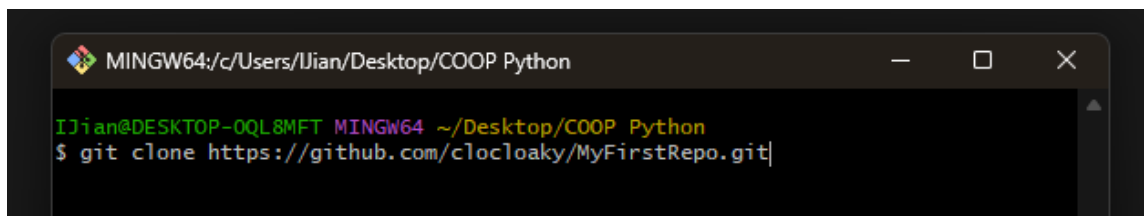
## Step 4: Clone the repository to your local machine

Now that you've created a new repository, you need to clone it to your local machine so you can work on it. To clone the repository, follow these steps:

1. Open Git Bash on your computer.



2. Navigate to the directory where you want to store your repository. (cmd line) Another option is you can right-click and open git bash in the folder you want to store your repository and it will automatically path to that folder.
3. Type the following command: `git clone https://github.com/your-username/your-repository.git` (replace "your-username" and "your-repository" with your actual username and repository name).



4. Press "Enter" to clone the repository to your local machine.

Name	Date modified	Type	Size
Lesson 1	9/14/2022 8:48 PM	File folder	
Lesson 2	9/28/2022 8:38 PM	File folder	
Lesson 3	10/4/2022 6:44 PM	File folder	
Lesson 4	9/12/2022 8:38 PM	File folder	
Lesson 5	9/12/2022 8:38 PM	File folder	
MyFirstRepo	2/28/2023 8:00 PM	File folder	

Notice how there is a new folder here named after my repo :)

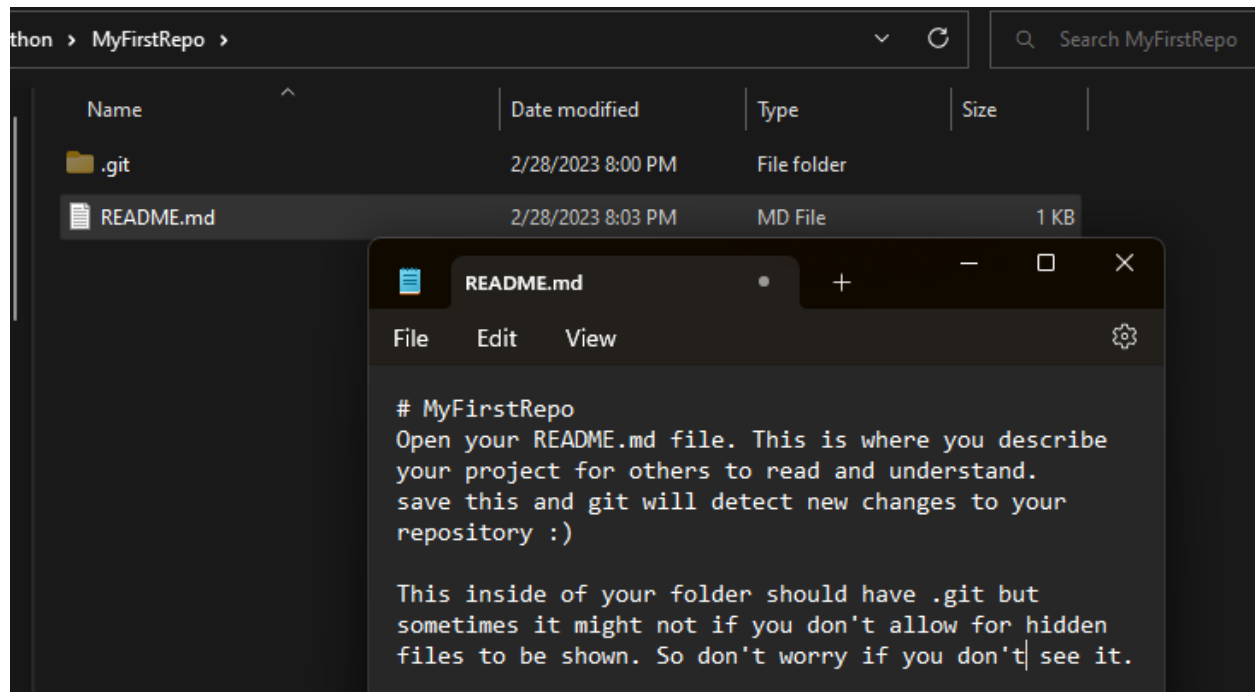
```
MINGW64:/c/Users/IJian/Desktop/COOP Python
IJian@DESKTOP-OQL8MFT MINGW64 ~/Desktop/COOP Python
$ git clone https://github.com/clocloaky/MyFirstRepo.git
Cloning into 'MyFirstRepo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

IJian@DESKTOP-OQL8MFT MINGW64 ~/Desktop/COOP Python
$
```



## Step 5: Make changes to the repository

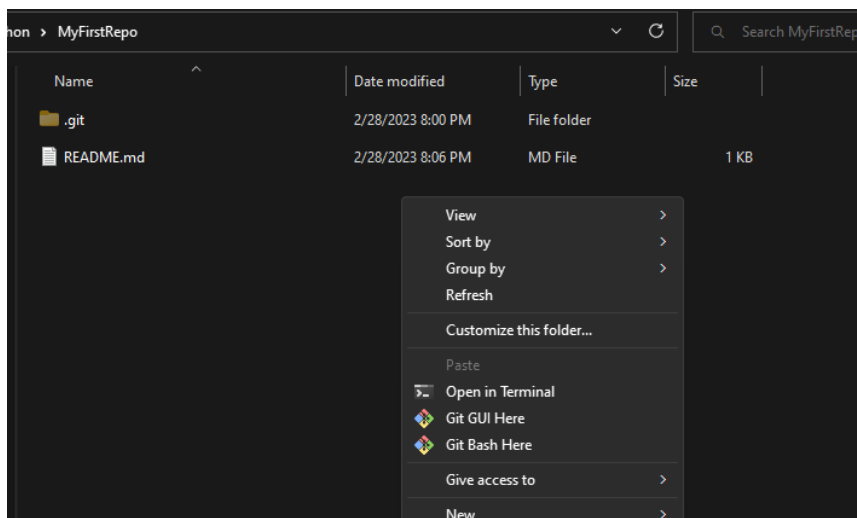
Now that you have a copy of the repository on your local machine, you can make changes to it. You can add new files, modify existing files, and delete files as needed.



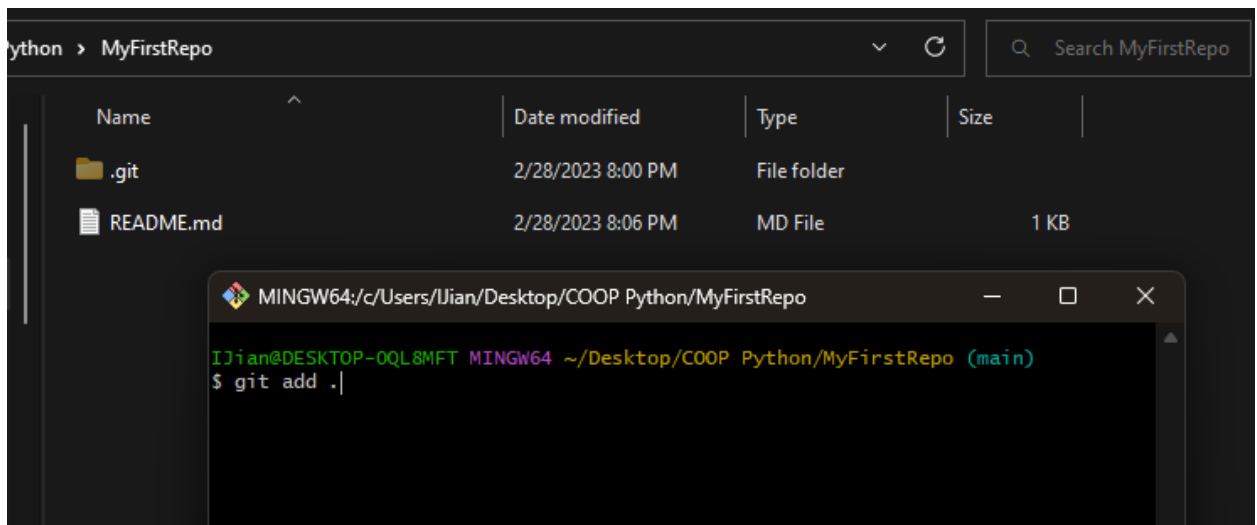
## Step 6: Commit your changes

Once you've made changes to the repository, you need to commit them so that they're saved in Git's version control system. To commit your changes, follow these steps:

1. Open Git Bash on your computer.
2. Navigate to the directory where your repository is stored.



3. Type the following command: `git add .` (the period at the end is important).

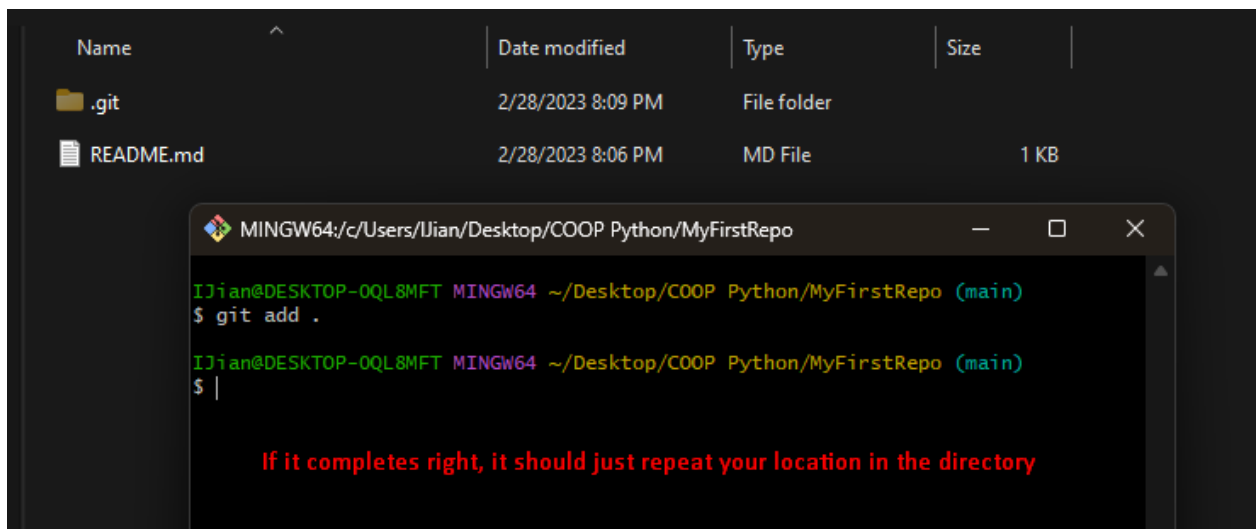


The screenshot shows a file explorer window for a directory named 'MyFirstRepo'. It contains a table with the following data:

Name	Date modified	Type	Size
.git	2/28/2023 8:00 PM	File folder	
README.md	2/28/2023 8:06 PM	MD File	1 KB

Below the file explorer is a terminal window titled 'MINGW64: c:/Users/IJian/Desktop/COOP Python/MyFirstRepo'. The terminal shows the following commands and output:

```
IJian@DESKTOP-OQL8MFT MINGW64 ~/Desktop/COOP Python/MyFirstRepo (main)
$ git add .
```



The screenshot shows the same file explorer window as above. The terminal window now shows the following commands and output:

```
IJian@DESKTOP-OQL8MFT MINGW64 ~/Desktop/COOP Python/MyFirstRepo (main)
$ git add .

IJian@DESKTOP-OQL8MFT MINGW64 ~/Desktop/COOP Python/MyFirstRepo (main)
$
```

Below the terminal output, there is a red text overlay that reads: "If it completes right, it should just repeat your location in the directory".



4. Type the following command: `git commit -m "Your commit message here"` (replace "Your commit message here" with a brief description of the changes you made).

Name	Date modified	Type	Size
.git	2/28/2023 8:09 PM	File folder	
README.md	2/28/2023 8:06 PM	MD File	1 KB

```
MINGW64/c/Users/IJian/Desktop/COOP Python/MyFirstRepo
IJian@DESKTOP-OQL8MFT MINGW64 ~/Desktop/COOP Python/MyFirstRepo (main)
$ git add .
IJian@DESKTOP-OQL8MFT MINGW64 ~/Desktop/COOP Python/MyFirstRepo (main)
$ git commit -m "This is my first commit message :)"
```

5. Press "Enter" to commit your changes to Git.

Name	Date modified	Type	Size
.git	2/28/2023 8:10 PM	File folder	
README.md	2/28/2023 8:06 PM	MD File	1 KB

```
MINGW64/c/Users/IJian/Desktop/COOP Python/MyFirstRepo
IJian@DESKTOP-OQL8MFT MINGW64 ~/Desktop/COOP Python/MyFirstRepo (main)
$ git add .
IJian@DESKTOP-OQL8MFT MINGW64 ~/Desktop/COOP Python/MyFirstRepo (main)
$ git commit -m "This is my first commit message :)"
[main 7bffaabd] This is my first commit message :)
1 file changed, 5 insertions(+), 1 deletion(-)
IJian@DESKTOP-OQL8MFT MINGW64 ~/Desktop/COOP Python/MyFirstRepo (main)
$ |
  Your cmd line will tell you all of the changes it is committing to the branch
```

6. Now hol' up. Git saves your repository locally onto your computer (local version control). You will still need to push it to GitHub.

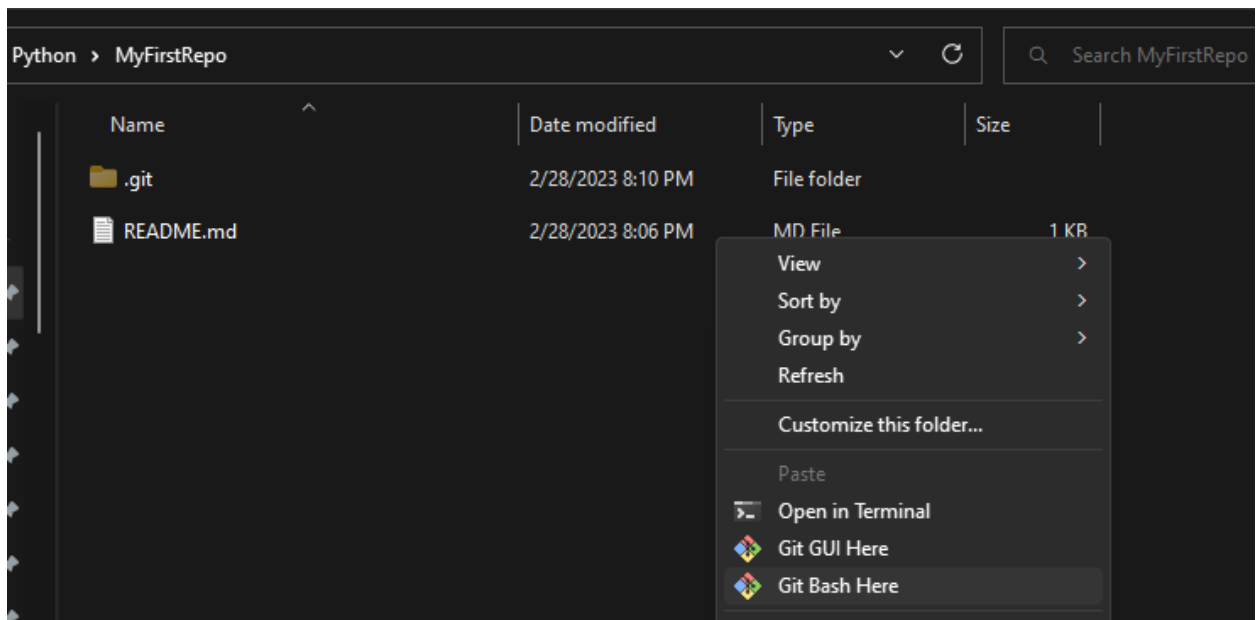
## Step 7: Push your changes to GitHub

Now that you've committed your changes to Git, you need to push them to GitHub so that they're saved in the online repository. To push your changes to GitHub, follow these steps:

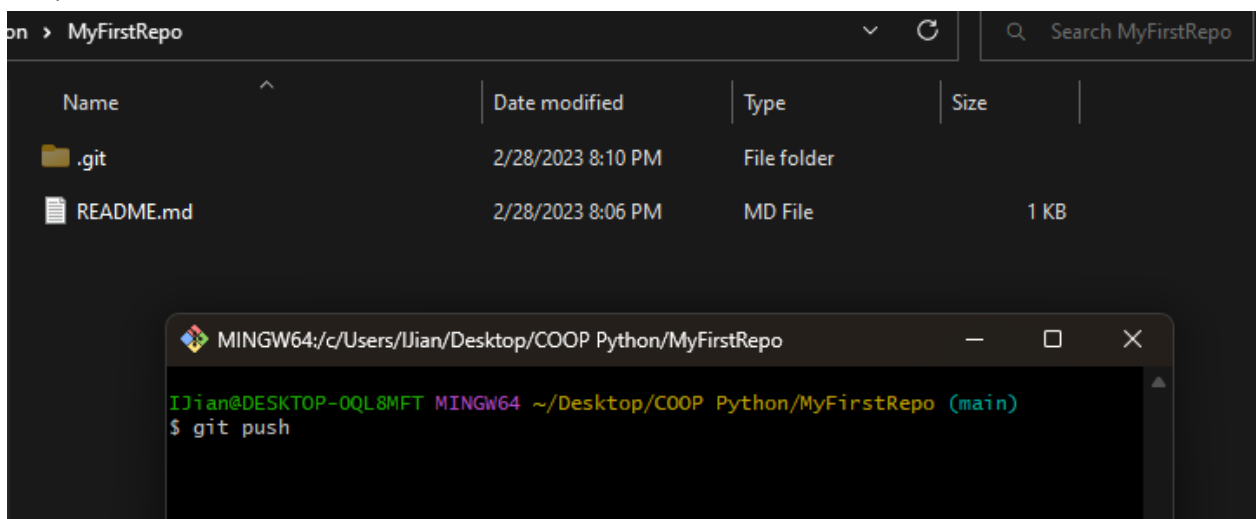
1. Open Git Bash on your computer.



2. Navigate to the directory where your repository is stored.

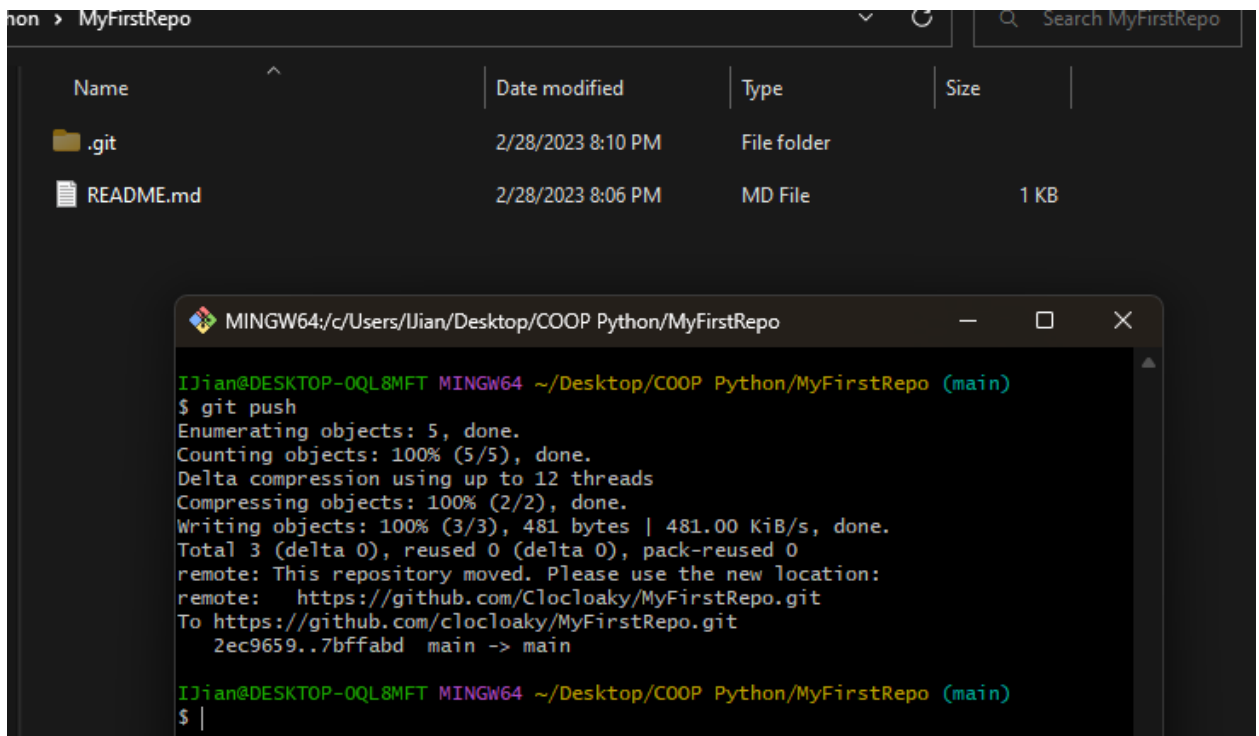


3. Type the following command: git push (you may be prompted to enter your GitHub username and password).





4. Press "Enter" to push your changes to GitHub.



The screenshot shows a file explorer window titled "MyFirstRepo" with a search bar. It contains a table with the following data:

Name	Date modified	Type	Size
.git	2/28/2023 8:10 PM	File folder	
README.md	2/28/2023 8:06 PM	MD File	1 KB

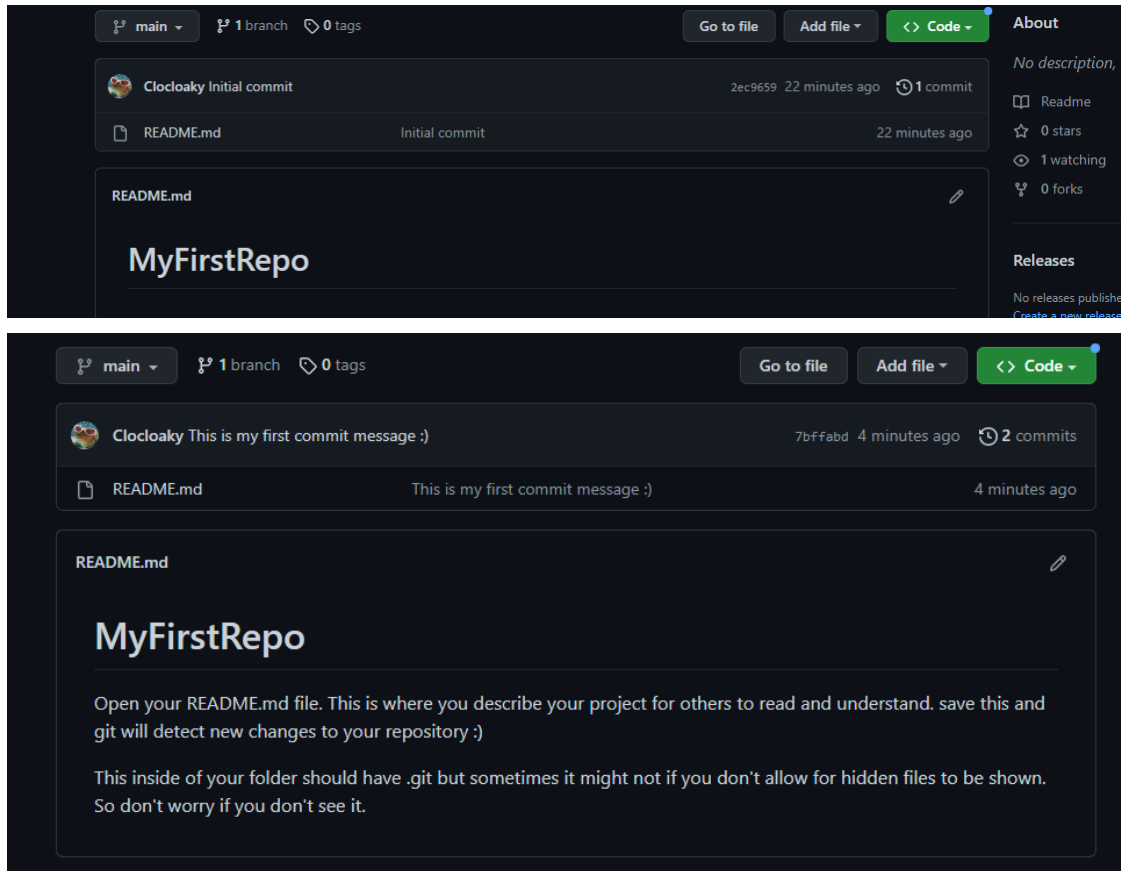
Below the file explorer is a terminal window titled "MINGW64: c:/Users/IJian/Desktop/COOP Python/MyFirstRepo". The terminal shows the following output for the command `git push`:

```
IJian@DESKTOP-0QL8MFT MINGW64 ~/Desktop/COOP Python/MyFirstRepo (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 481 bytes | 481.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: This repository moved. Please use the new location:
remote: https://github.com/Clocloaky/MyFirstRepo.git
To https://github.com/clocloaky/MyFirstRepo.git
2ec9659..7bffa9d main -> main

IJian@DESKTOP-0QL8MFT MINGW64 ~/Desktop/COOP Python/MyFirstRepo (main)
$ |
```



5. Check your GitHub...



And that's it! You've successfully set up a repository on GitHub, cloned it to your local machine, made changes to it, committed your changes, and pushed your changes to GitHub. You can now continue to work on your repository and use Git to manage your code changes.

