MRM TASK 1

Created	@December 20, 2021 10:11 PM
Tags	
Created by	

GIT AND GITHUB

What is GIT?

Git is a version control system which lets us keep track of changes and modifications that we make to a file over time. With GIT, we can have a sort of “undo” functionality to our files like the time machine on MacOS. We can work on copies of our files without changing the original file and then merge them afterwards. For example, I am working on revamping the MRM website, but at the same time it would not be very safe to start altering its components because it might break some function or the other on the website. With GIT, I can clone the code of the website and play around with the website’s code. Then when everyone in the team is satisfied with the changes I made to the website, we can merge the clone to the original file.

But GIT is not just useful for developers and programmers, it can also be used by designers, photographers etc. to keep track of their text files, documents and images.

Installing GIT

Like any other piece of software we need to install GIT on our computers, It can be installed via its official website.

Configuring GIT

To check and verify that GIT is installed on cmd run this command:

```
git --version
```

The next thing you'll need to do is to set your username and email address. Git will use this information to identify who made specific changes to files. To set your

username, type and execute these commands:

```
git config --global user.name "floofy_244"
git config --global user.email "iamaviralmalik@gmail.com"
```

How to Create and Initialize a Project in GIT

I have created a folder on my desktop called MRM Tasks. Using the command line, navigate to your new project's location. For me, I would run the following commands:

```
cd desktop
cd MRM Tasks
```

Now to initialize your project, simply run `git init`. This will tell Git to get ready to start watching your files for every change that occurs. It looks like this:

The first line is the path to where the folder exists. The second line is the command `git init`, and the third line is the response sent back telling me that the repository has been initialized.

What is GitHub?

GitHub is an online hosting service for Git repositories. With all our projects hosted on GitHub we can work on them remotely without having access to the computer on which we created the project. You can work and make changes to your projects even while travelling and have the latest version of all your files easily accessible to me anywhere in the world.

GitHub lets you store your repo on their platform. Another awesome feature that comes with GitHub is the ability to collaborate with other developers from any location.

after I had created and initialized my project locally on my laptop I pushed the repository to GitHub.

How to push a repo to GitHub

First we need to create an account on GitHub. Then click on the `+` symbol on the top right corner of the page then choose "New repository". Give your repo a name then scroll down and click on "Create repository".

Different States of a file

Committed State: A file is in the **committed** state when all the changes made to the file have been saved in the local repo. Files in the committed stage are files ready to be pushed to GitHub.

Modified State: A file in the **modified** state has some changes made to it but it's not yet saved. This means that the state of the file has been altered from its previous state in the committed state.

Staged State: A file in the **staged** state means it is ready to be committed. In this state, all necessary changes have been made so the next step is to move the file to the commit state.

Adding files in Git

Initialized repos are not tracked by Git as it is. To start tracking we need to add files to git using the command: `git add .`

the dot here refers to all the files inside the repo. To add only a specific file we can use `git add index.html`

Now the project is in the staged state. You will not get a response after this command, but to know what state your file is in, you can run the

`git status` command

Committing Files

After the staged state is the committed state, to commit the files we use

`git commit -m “first commit”`.

The first part of the command `git commit` tells Git that all the files staged are ready to be committed so it is time to take a snapshot. The second part `m "first commit"` is the commit message. `-m` is shorthand for message while the text inside the quotes is the commit message.

```
avira@Aviral-laptop MINGW64 ~/desktop/MRM (main)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

avira@Aviral-laptop MINGW64 ~/desktop/MRM (main)
$ git commit -m "first commit"
[main (root-commit) 9e8ca42] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

Push the repo to GitHub

after creating the repo, GitHub prompts us to create a repo locally or push and existing one.

In my case, the repo already exists locally so we will only push the repo locally using the commands:

```
avira@Aviral-laptop MINGW64 ~/desktop/MRM (main)
$ git branch -M main

avira@Aviral-laptop MINGW64 ~/desktop/MRM (main)
$ git remote add origin https://github.com/floofy244/MRM-Tasks.git

avira@Aviral-laptop MINGW64 ~/desktop/MRM (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 225 bytes | 225.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/floofy244/MRM-Tasks.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

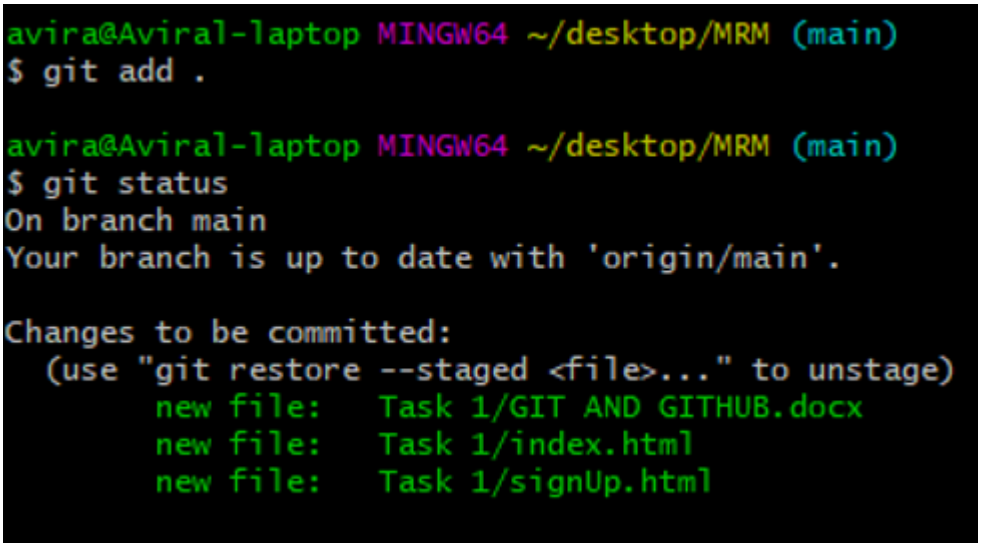
The first command `git remote add origin https://github.com/floofy244/MRM-Tasks.git` creates a connection between your local repo and the remote repo on GitHub.

The URL for your remote project should be entirely different from the one above. So to follow along, make sure you are following the steps and working with your own remote repo. You won't usually get a response after executing this command but make sure you have an internet connection.

The second command `git branch -M main` changes your main branch's name to "main". The default branch might be created as "master", but "main" is the standard name for this repo now. There is usually no response here.

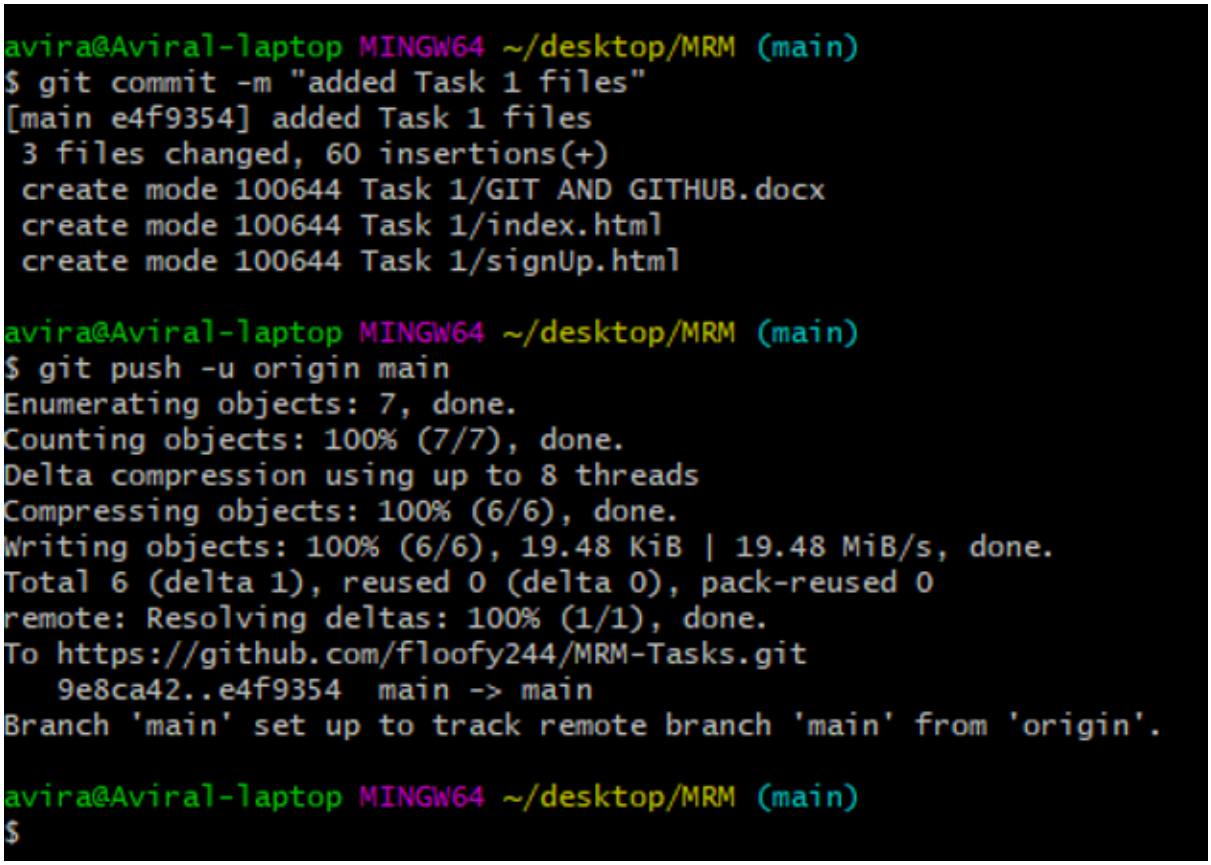
The last command `git push -u origin main` pushes your repo from your local device to GitHub.

Also our files were in the committed state. Now I made changes to the file and check the states.



Now we are going to add (stage) this file and then commit and push it. This is the same as we did previously.

We first add the file by using `git add .` which adds all the files in the folder (one file in our case). Then we commit the file by running `git commit -m "added new task"` followed by `git push -u origin main`.



Using Branches in GIT