**Git**

To setup for first time:

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/ <git id> / <git repository>

git remote set-url origin https://< personal access token >@github.com/<git id>/<git repository>

git push -u origin main

Text

Description automatically generated

Explanation:

On the terminal go to the folder that needs to be made the repository,

* git init – specifies that this folder is the repository
* git commit – commits local changes to stagging area
* git branch – to create a different branch called main (else everything will be committed to master)
* git remote – specifies remote linkages
* git remote add origin <URL path to github> - specifies destination github path
* git remote set-url origin https://< personal access token >@github.com/<git id>/<git repository>

Creating a personal access token helps identify you (replaces the earlier SSH key for authentication)

To create a new personal access token :

Settings -> Developer Settings -> Personal access tokens -> Generate new token -> specify all privileges (select all if not sure) -> create -> copy the token

Graphical user interface, text, application, website, Teams

Description automatically generated

* git push -u origin main

git push -u <destination> <local branch or master>

Output:

A screenshot of a computer screen

Description automatically generated with medium confidence

Now, to create a sub folder inside the repository:

Just copy paste the folder into the local repository, then follow these steps:

1. git init
2. git add .
3. git commit -m “message/comment”
4. git push -u origin <branch-name/master>

Now, to create a submodule:

Note: sub folder is different from submodule.

Submodule is a nested repository – which means that this folder will also have a .git folder inside it.

Reference:

[1] <https://devconnected.com/how-to-add-and-update-git-submodules/>

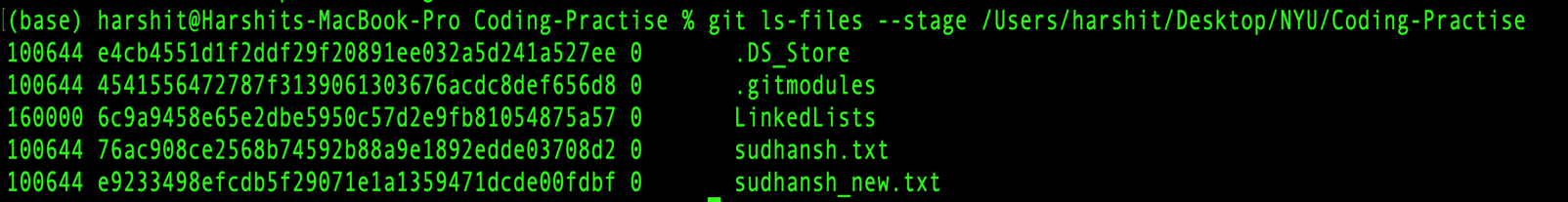
[2] <https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/The-git-submodule-add-example>

[3] <https://stackoverflow.com/questions/12898278/issue-with-adding-common-code-as-git-submodule-already-exists-in-the-index>

To view files already staged locally:

* git ls-files –stage <local repository path>

Example:



More Info:

<https://stackoverflow.com/questions/12898278/issue-with-adding-common-code-as-git-submodule-already-exists-in-the-index>

To remove folder from staging area:

* git rm --cached projectfolder

Example:

Graphical user interface, text, application

Description automatically generated

Was trying to add a sub-module – didn’t work – for future reference:

(Note a sub module is also a repository – so you have to follow all the steps of committing a file to that sub-repository i.e. sub-repository would need git init to initialize the repo, etc. it behaves exactly like a normal repo.

Difference? After you commit changes inside the sub-repo, do “cd ..” to come to the parent repo, then :

* Git init
* Git add . [to add the sub-module to the parent repo]
* Git commit

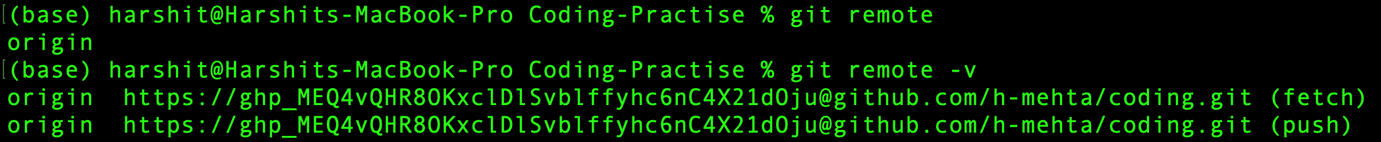
Before doing git commit, one can check if the sub-module i.e. sub-repo has been added correctly or not by doing “git status”; if the sub-module has been added then you will see .gitmodules – this lists all the sub-modules in the parent repo)

Graphical user interface, text, application

Description automatically generated

To check the remote linkages:

* Remote linkages basically tell you what remote-repository will the code get pushed to



**Git Remote :**

Git remote is a command-line utility that allows you to manage remote repositories that your local repository interacts with. A remote repository is a version of your repository hosted on a remote server, typically on services like GitHub, GitLab, or Bitbucket. You can use git remote to interact with these remote repositories and perform various tasks, such as:

1. Adding a remote: You can use git remote add to add a new remote repository to your local repository.

git remote add origin [git@github.com:username/repo.git](mailto:git@github.com:username/repo.git)

1. Listing remotes: You can use git remote or git remote -v to list the existing remotes for your local repository.

git remote -v

origin git@github.com:username/repo.git (fetch)

origin git@github.com:username/repo.git (push)

1. Removing a remote: You can use git remote remove to remove a remote repository from your local repository.

git remote remove origin

1. Renaming a remote: You can use git remote rename to rename an existing remote.

git remote rename origin upstream

1. Fetching changes from a remote: You can use git fetch to retrieve changes from a remote repository.

git fetch upstream

1. Pushing changes to a remote: You can use git push to send changes to a remote repository.

git push origin main

Useful Links:

**Error**

[1] “Remote origin already exists”

<https://www.cloudbees.com/blog/remote-origin-already-exists-error>

[2] submodule related: <https://stackoverflow.com/questions/62056294/github-folders-have-a-white-arrow-on-them>