

Open a GitHub Account

Create New Repository

Git Terms:

Working Directory

Local Repository

Remote Repo

*Conversational Commit Message

Git Global Commands (এই গুলো Git Bash Command)

`git config --list` → check git configuration

`git config --global user.name “user_name”` → set username in local machine

`git config --global user.email “email”` → set email in local machine

`pwd` → shows present working directory

`touch file_name` → shows present working directory

`git status` → check git status

`ls` → show files, folders insider the directory

`ls -a` → show all (including hidden) files, folders insider the directory

`git add file_name` → tell git to track the specific *file*

`git add --all` → tell git to track all folders and subfolders from the root to core

`git add .` → tell git to track all folders and subfolders of current directory (similar to above one)

`git log` → shows the log of git commits

`git log --oneline` → shows shorter logs

`git reset --hard commit_id` → undo specific commit

`git reflog` → view total logs including reset and others

`git rm file_name` → shows all the available branches

Branching Concepts:

`git branch --list` → shows all the available branches

`git branch dev/add-heading-text` → create a branch here
dev/add-heading-text is a naming convention.

`git switch dev/add-heading-text` → এই ব্রাঞ্চে switch হয়ে যাবে।

`git merge dev/add-heading-text` → এখানে আমরা আগে main/master ব্রাঞ্চে গিয়ে তারপর সেখান থেকে dev/add-heading-text এই changes গুলোকে main branch এ টেনে আনছি। এই সময়ে git reflog করলে এমন দেখাবেঃ

```
8054889 (HEAD -> master, dev/add-heading-text) HEAD@{0}: merge dev/add-heading-text: Fast-forward
```

এখানে কিন্তু কমা দিয়ে বুঝিয়ে দেয়া হচ্ছে যে কোন branch এর সাথে merge টা করা হয়েছে।

`git branch -d branch_name` → *branch_name* নামের ব্রাঞ্চটা ডিলিট করার জন্য ব্যবহৃত হয়। তবে, ছোট হাতের d দিয়ে করলে warning পাবো

`git branch -D branch_name` → বড় হাতের D দিয়ে করলে warning ছাড়াই branch টা delete হয়ে যাবে। (unsafe)

`git branch -m branch_name` → modify (change) branch name. আর এটা করতে হলে যে ব্রাঞ্চে নাম চেইঞ্জ করবো আগে ঐ ব্রাঞ্চে switch করে নিতে হবে।

যদি এমন হয়, commit না করেই আমরা চাই এক branch থেকে অন্য একটা ব্রাঞ্চে move করতে। কিন্তু আমরা আমাদের Work In Progress (WIP) কে পরে ব্যবহার করতে চাই, সেক্ষেত্রে আমরা stash করি।

stash করা মানে হচ্ছে, সেটা local directory সাময়িকভাবে রাখা বা বলা যেতে পারে temp আকারে রাখা। পরে সেখান থেকে আমরা সেটাকে commit করে নিতে পারি।

`git stash` → for creating stash on the current directory of the WIP.

`git stash --list` → for listing the stashes.

`git stash show -p` → shows what was stashed

`git stash pop` → grab the latest shash from the list

`git stash apply stash_id` → grab the latest shash from the list

Copy a previous branch to a new branch:

প্রথমে যে branch কে copy করতে চাই ঐ brach এ যেতে হবে। এরপর সেখানে থাকা অবস্থায়ঃ

`git checkout -b branch_name` → copy a previous branch to a new branch

e.g: `git checkout -b 3.20` → এখন

gitignore

কোন একটি file কে ট্রাক না করার জন্য এই ফাইলের মাঝে ঐ ফাইলের নাম লিখে দিলেই সেটা করবে। অর্থাৎ git আর সেই ফাইলকে ট্রাক করবে না।

● কিন্তু যদি আমরা কোন ফাইলকে একবার commit করেই ফেলি, কিন্তু এরপরে গিয়ে আমরা চাইলাম যে সেটাকে ignore করতে, সেক্ষেত্রে হবেঃ

`git rm --cached file_or_folder_name` → removes the file or folder from the current tracking system but will keep it in the working directory.
অর্থাৎ ফাইলটাকে git পুরোপুরি delete করে দেবে না, কিন্তু সেটাকে আর track করবে না।

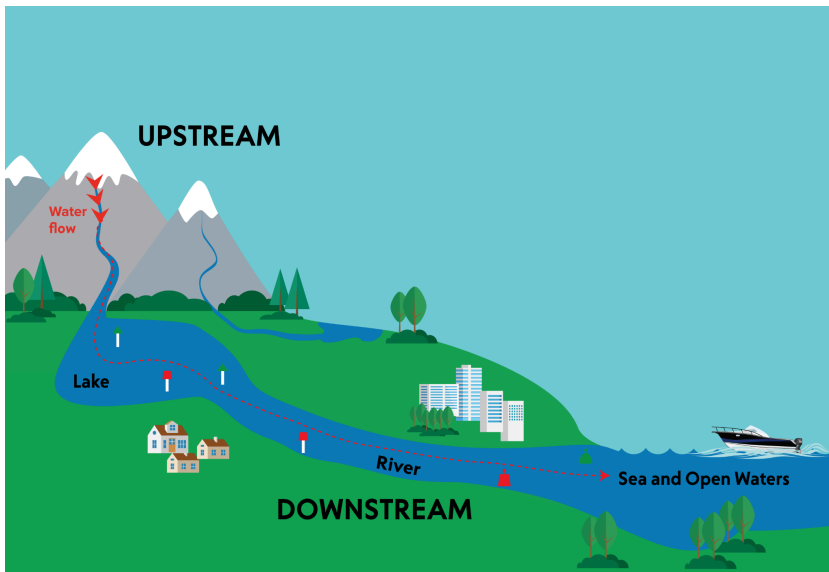
`git rm --cached file_or_folder_name` → removes the file or folder
`git diff file_or_folder_name` → show the changes between commits

`git pull` → pulls the latest changes from the remote Repo. এখানে মনে রাখতে হবে, Remote যে ব্রাঞ্চে changes করা হয়েছে, locally ও সেই ব্রাঞ্চে গিয়েই pull request টা করতে হবে।

Forking:

Upstream → যে account থেকে fork করা হয়। upstream মনে রাখার জন্য ধরে নিবো আমরা ঐ repo account থেকে আমাদের account এ code বারে বারে আমাদের account এ এসে পড়ছে। 😊

Forked Account → আমার নিজের account / যে account এ ফোর্ক করা হয়েছে।



Cloning a Repo:

- HTTPS Process: github এ গিয়ে সেখান থেকে https এর URL টা কপি করে নিয়ে এসে `git clone https_url` এই command দিয়ে দিলেই হয়ে যাবে।
- SSH Process: github এ গিয়ে সেখান থেকে

● How to maintain 2 origins at the same time:

```
git init
git add .
git commit -m "First commit"
git branch -M main
git remote add vercel YOUR_PERSONAL_ACCOUNT_REPOSITORY_URL
git remote add lwscod LWS_CLASSROOM_REPOSITORY_URL
git remote (optional. for checking repositories)
git push vercel main
git push lwscod main
```

Command Line Commands:

cd = change directory

cd.. = 1 folder back

cd [folder name] = 1 folder forward

Tab button = auto complete the tab

GitHub Commands

```
echo "# my-first-repo" >> README.md
```

```
git init
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin
```

```
https://github.com/Rakibul-Hashan/my-first-repo.git
```

```
git push -u origin main
```

Step:02 Run

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"
```

Update a repository:

```
git add .
```

```
git commit -m "islamic img added"
```

git push

git remote add origin <https://github.com/Rakibul-Hashan/my-first-repo.git>

git branch -M main

git push -u origin main

Common Mistakes related to github:

1. Using different name of the html file other than *index.html*
2. Remote origin already exists:

git remote set-url origin ["repo url"]

```
git remote add origin
```

```
https://github.com/Rakibul-Hashan/simple-ecommerce-site-with-auth.git
```

এখানে, “add” এর বদলে set-url করে দিলেই origin new repo তে সেট হয়ে যাবে।

How to open new branch and merge them for team work:

Link: <https://github.com/joshnh/Git-Commands>

Use this for checking the current branch status/ name:

git branch

For creating a new branch, Here *image-add* is the branch name:

git branch image-add

Use this for Create a new branch:

```
git checkout [branch name]
```

Then add the repository again:

```
git add .
```

Push it again. But here we will get an error.

```
[fatal: The current branch image-add has no upstream branch.  
To push the current branch and set the remote as upstream, use  
]
```

```
git push
```

To solve the error we have to use this:

```
git push --set-upstream origin image-add
```

Now use again

```
git push
```

নতুন [branch name] এ সব কিছু চলে গেছে। এখন এডারে merge করতে হবে। এজন্য আমাদের পুনরায় *main* branch এ গিয়ে *merge* করতে হবে
সো, এজন্য দুইটা স্টেপঃ

```
git checkout main
```

```
git merge image-add
```


Git এ কাজ করার জন্য কিছু Good Practice.

1. Main branch এ কাজ করা যাবেনা, Feature branch এ কাজ করতে হবে।
2. আগেই main branch এর কোড *pull* করে নিতে হবে
3. এরপর কী কী branch আছে সেগুলো দেখে নিতে হবে [[git status](#)]
4. তারপর আমাদের জন্য নিউ branch বানিয়ে সেখানে checkout করে সেখানে চলে যেতে হবে [\[git checkout -b \[branch name\] \]](#)
5. এর পর আমরা আমাদের ফাইলে কিছু change করে তারপর আমরা চাইলে আমাদের কোন ফাইলে করেছি সেটা দেখতে পারবো [git status](#) দিয়ে. তারপর আগের মতই [git add](#) . এরপর [git commit -m "message"](#) -----> [git push](#) দিতে হবে।
6. এরপর আবার main branch এ নিতে চাইলে আবারও সেই একই ভাবে, বর্তমান branch → main branch → আগের মত *merge* করতে হবে।

Git Conflict:

যখন একই change দুইটা branch থেকে আসবে এবং [git merge](#) করার সময় conflict হবে তখন সেখানে অপশন পাওয়া যাবে। সেটা দেখে ডিসিশন নিতে হবে যে, কোনটা আমরা রাখতে চাই।

Transfer Private repo to Public


```
git clone --bare <repo.link>
```

```
git push --mirror <public.repo.link>
```

```
git remote -v
```

```
git remote set-url origin <public.repo.link>
```

Reference Youtube Video in Bangla -

 Git/GitHub Refresher - Reactive Accelerator course by Learn with Su...

My thanks goes to for sharing such document with us.

@rakibulhashanrabbi (youtube handle)

41:45 Git Branch

47:42 Merge

51:52 Merge CONFLICT handling

59:22 Stash

1:03:40 Git ignore

1:06:35 ignore a file that has already been committed

1:10:35 Github Account & Remote Repo

1:13:21 Create a new repo

1:16:54 Pull Request

1:22:23 Forking

1:26:26 Tips about Readme file

1:27:30 Clone a Repo (HTTPS)

1:32:04 Clone a Repo (SSH)

1:37:55 Tips

This file modified by

Md. Alamin,

Programmer

Java | Spring Boot | MySQL

<https://github.com/alamin5G/>