

**#01**

**Step 01: Software install-**

Website: <https://git-scm.com/downloads>

Gut bash terminal Check : git --version

**Step 2: How to Setup “GitHub” -**

Github account singup:

**#02**

**Step 01: Configuration -**

Terminal clear: **Ctrl+L**

```
git config --global user.name "ShahriarTBC" (Enter)
git config --global user.email "abc@gmail.com" (Enter)
git config --global user.name (show only user name)
git config --list (all configuration check)
```

**Step 02: Create Git Folder & Add Files -**

**Working Directory → Staging Area → Local Repository → Remote/Github**

Working Directory → 1st Stage (Untracked Stage)

Staging Area → 2nd Stage (No commits yet)

Local Repository → 3rd Stage

Remote/Github → 4th Stage

**#03**

**Make Working Directory(Create Folder) -**

mkdir first-repo	Create Folder : first-repo
cd first-repo	Go next folder
git init	Active repo (initialization repository)
ls -a	Hidden folder (.git) view
ls	All file /folder view
touch one.html	Create File : one.html
git status	Check project level status

```
D:           | Go fixed D drive (TAB key suggest) |
cd ..        | Go back previous folder |
pwd          | Present directory show (/c/User/Shahriar) |
rm -rf folder_name | Deleted any folder |
```

#05

### Working Directory → Staging area : Go 2nd Stage

```
git add File_Name    | Specific one file go Staging Area |
git add .            | All file Working D..→ Staging Area |
git diff            | File contains change/update view |
git restore File_Name | File contain previous condition |
git rm --cached     | Staging Area → Working Directory |
```

#06

### Staging Area → Local Repository : Go 3rd Stage

```
git commit -m“message” | Staging A.. → Local Repo.. |
git log                 | Commit full history show |
git add . && git commit -m“message” |Working → Local |
```

#### \*Last update HEAD file possible:

```
git reset --soft HEAD^   | Local Repository→ Staging Area |
git reset HEAD^         | Local Repository→ Working Directory |
git reset --hard HEAD^  | File Delete |
```

#07

### <---- All Revision ----->

#08

### Commit & HEAD & Undo

```
git log             | Commit full history show |
```

```
git log --oneline      | Key ID + Commit message |
git show               | A to Z full history (Only Head) |
git show commit_id/git show | Specific commit history |
git show HEAD~number   | Single commit history |
```

**\*Undo commit**

```
git checkout one.txt
| New content text remove |
```

```
git checkout previous_file_ID (git log --oneline)
|3rd commit → 1st commit|
| File gulake ager obosthay niye jabe |
```

```
git checkout master
|1st commit → 3rd commit (HEAD)|
| File ke ager obostha theke porer obosthay niye jabe |
```

**#09**

**<---- Commit Revision ---->**

**#10**

**GitHub Profile & Readme**

**#11**

**GitHub Profile design:** [Website Link](#)

**#12**

**GitHub Profile design:** [Website Link](#)

**#13**

## Repository or commit-git & github

Create a repository and upload any file.

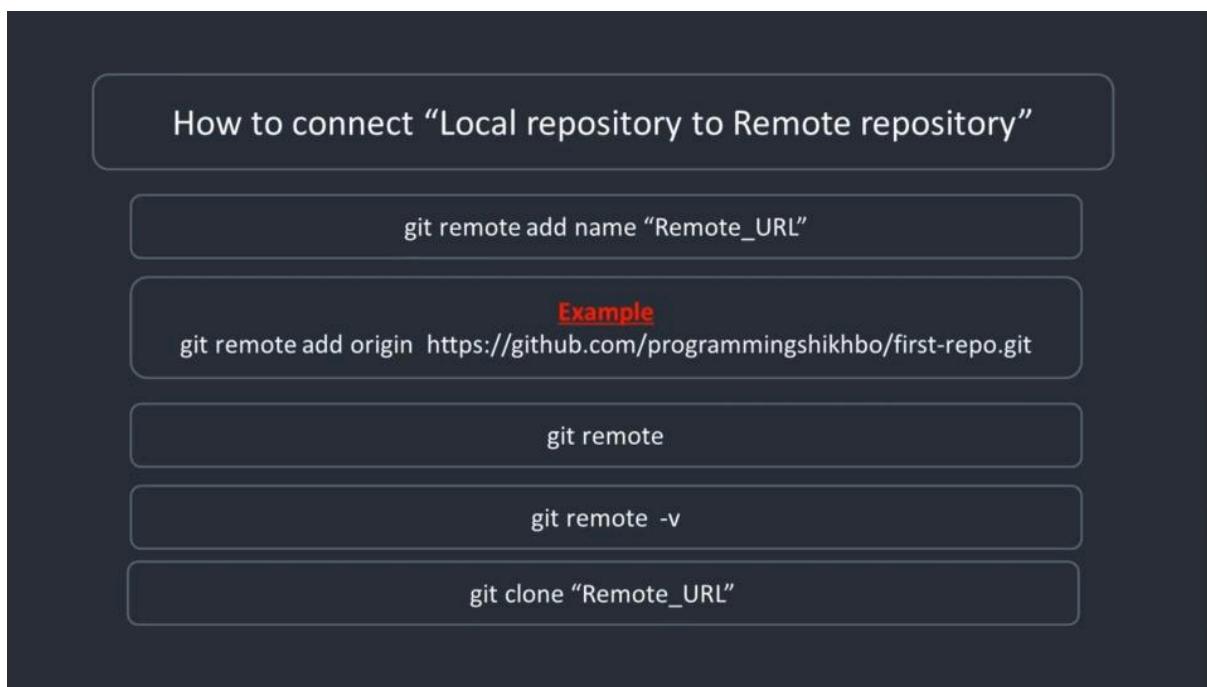
#14

### How to connect “Local repository to Remote repository”

Local repository: offline store (Example: Local Computer)

Remote repository: Online store (Example: GitHub)

```
git remote      | Check Connection (“Name”) |
git remote add origin repo_url | Go repo → Code → https://.. |
git remote -v | URL connection check |
git clone repo_url | GitHub to Desktop copy |
rm -rf folder_name | Deleted any folder |
```



#15

### Clone GitHub any project

```
git clone repo_url | GitHub to Desktop copy |
```

#16

### **Push the project to GitHub with VS Code**

GitHub Token : Setting--> Developer settings--> Personal access tokens--> Tokens (classic)--> Generate new token (classic)  
For general use--> (Note, repo, admin:org, admin:public\_key)

Token: ghp\_g6n3SBvMRDEs7moKcbIE3W42hhlnz017laHM

1. git remote add origin https://github.com/ShahriarTbc/First\_Tailwind\_Project.git
2. git remote set-url origin  
[https://ghp\\_g6n3SBvMRDEs7moKcbIE3W42hhlnz017laHM@github.com/ShahriarTbc/First\\_Tailwind\\_Project.git](https://ghp_g6n3SBvMRDEs7moKcbIE3W42hhlnz017laHM@github.com/ShahriarTbc/First_Tailwind_Project.git)

1. git init
2. git status
3. git add .
4. git commit -m "first commit"
5. git branch -M main
6. git push -u origin main

- 5.1. git remote add origin https://github.com/ShahriarTbc/<--\*your site-link\*-->
- 5.2. git remote set-url origin https://<--\*your token--\*>@github.com/ShahriarTbc/<--\*your site-link\*-->

README.md File add:  
git pull origin main --rebase  
git push origin main

#17

## **Push the project to GitHub with Git bash Terminal**

**#18**

### **Git Pull (Online update to Offline update)**

git pull | Github text content update to Offline update |

**#19**

### **Branching (Main/Master) and Merging**

GitHub create branch

**#20**

### **Branching (Main/Master) and Merging**

Git bash create branch

git branch  
| Check branch | (\*main → active branch)

git branch feature-1  
| Create feature-1 branch |

git checkout feature-1  
| Move feature-1 branch| (\*feature-1 → active branch)

git branch -d feature-1  
| Deleted feature-1 branch | (Before go Main it's work)

git push -u origin feature-2  
| Push feature-2 branch |

**#21**

## Merge Project

```
git checkout -b "feature-10"
| Create and Move branch |
```

### \*Do Merge

\*Go main branch\*

```
git merge feature-10 | feature-10 → Main branch |
```

#22

## Issues

GitHub issues project

#23

## Fast forward Merge (2-way)

Git school: [Website Link](#)

The screenshot shows a terminal window with the following content:

```
Free Explore
Local Repository
HEAD: master

$ git commit -m "second
commit"
$ git commit -m "third
commit"
$ git branch feature
$ git checkout feature
$ git commit -m "fourth
commit"
$ git commit -m "fifth
commit"
$ git commit -m "sixth
commit"
$ git checkout master
$ git merge feature
You have performed a fast-forward
merge.
$ enter git command
```

To the right of the terminal, a diagram illustrates the commit history and branches:

- Commits: first commit (e137a9b..), second commit (cbe2962..), third commit (28d0d0a..), fourth commit (cc79e25..), fifth commit (4319f54..), sixth commit (a91072d..).
- Branches:
  - feature (orange box)
  - master (orange box)
  - HEAD (green box)
- Relationship: The commits from 'feature' are shown as grey circles connected by arrows pointing left towards the 'master' branch, indicating a fast-forward merge.

#24

## Fast forward Merge (3-way)

clear | terminal clear |

The terminal window shows the following command history:

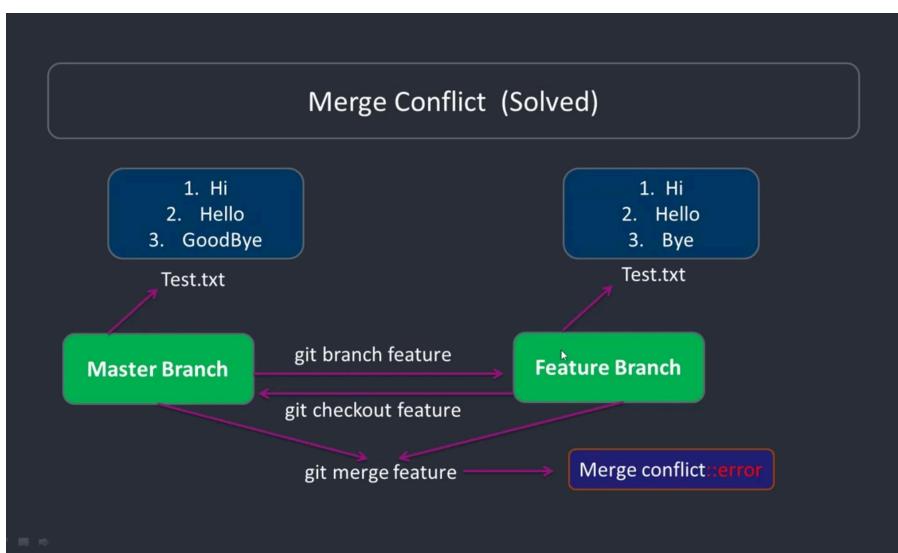
```
Have fun!
$ git commit -m "second commit"
$ git commit -m "third commit"
$ git branch feature
$ git checkout feature
$ git commit -m "fourth commit"
$ git commit -m "fifth commit"
$ git commit -m "sixth commit"
$ git checkout master
$ git commit -m "seventh commit"
$ git commit -m "eighth commit"
$ git commit -m "ninth commit"
$ git merge feature
$ git log
```

Below the terminal is a commit graph diagram. The master branch has 9 commits: first commit (e137e9b..), second commit (6f10284..), third commit (follow8..), fourth commit (8609fc2..), fifth commit (f292931..), sixth commit (42401a6..), seventh commit (298aade..), eighth commit (ff97f0b3..), and ninth commit (90f7012..). A green circle at the end of the master branch is labeled "HEAD" and "master". A separate "feature" branch has 3 commits: 1. Hi, 2. Hello, 3. GoodBye. A merge commit (colorer7c7..) is shown where the feature branch is merged into the master branch. This merge commit is labeled "Merge".

#25

### Merge complete error

Local server



Duita alada branch a same file er same line a alada text rekhe merge korle error show korbe. Tai jekono ekta file er modhe change korte hobe.

#26

Merge complete error solved

Online server, GitHub.