

Git & Github

(1) Setup :- Download git, github desktop.

- setup Github.com

Sign in into Github.com → Create New Acc.
→ Log in into old Acc.

Let username ⇒ ab17

Email ⇒ ab17@gmail.com

- setup git (open cmd OR Git bash)

\$ git --version → To check git is installed or not

\$ git config --global user.name "ab17"

\$ git config --global user.email "ab17@gmail.com"

\$ git config --list → To list git config

- setup Github Desktop

→ Open > Login using github acc.

→ Config > manual config

(2) Creating git repository

→ file explorer > folder (you want to create git repo)

> Right click > Open in Terminal

\$ git status → To check folder is a git repo or not

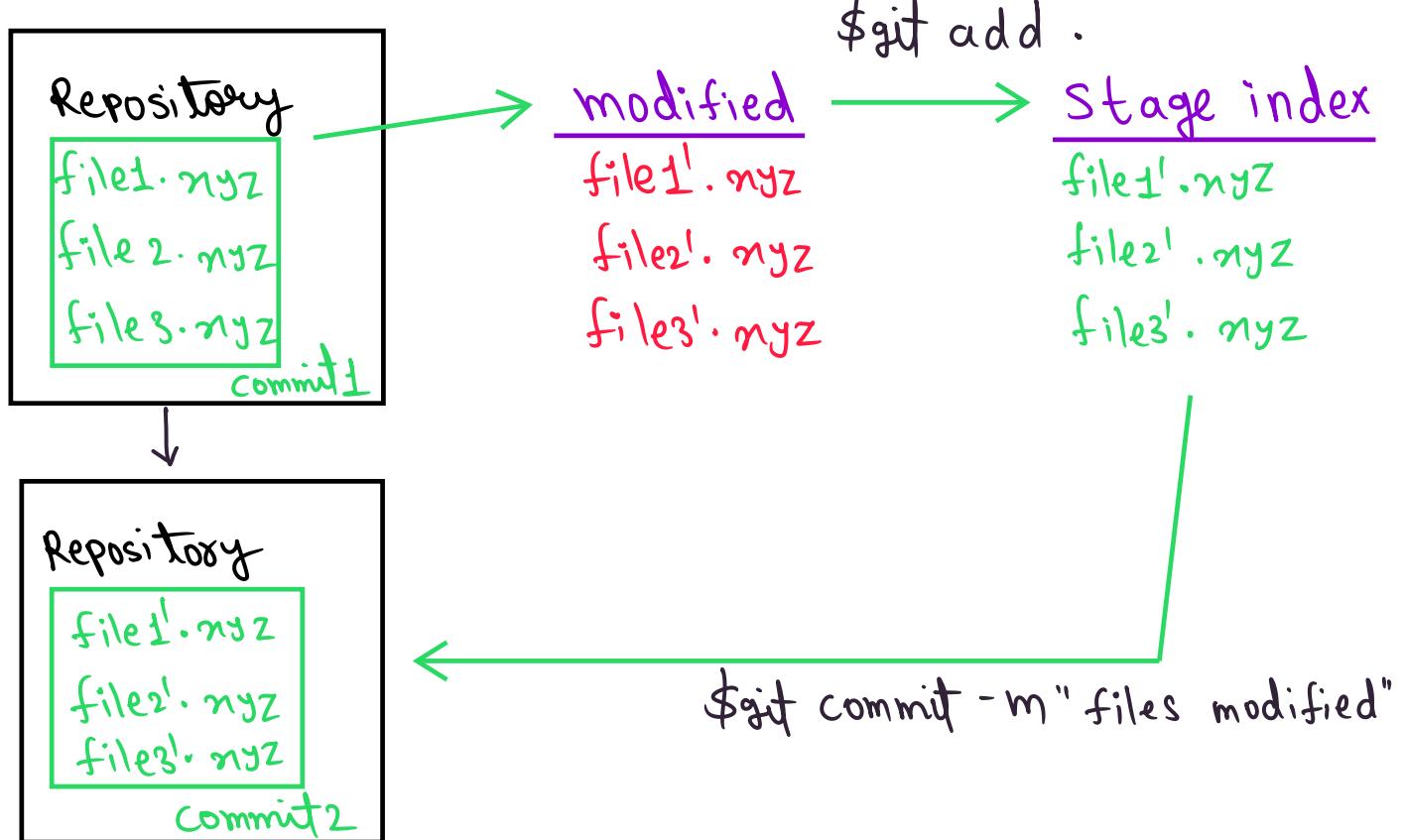
\$ git init → To initialize folder as git repo

(3) cloning remote github repo at local

- Go to the remote repo you want to clone
- Copy its https code (Code > Https code)
- Open Terminal at Local

\$git clone <https://github.com/username/repo.git>

(4) working with git repo



(5) View modifications

\$git status → To view modified files & folders
\$git diff file1.nyz → To view modifications of file1.nyz, where the file is present in working directory

\$git diff --cached file1.nyz → To view changes of file1.nyz, when it is staged.

(6) Doing commit

Let file1.nyz and file2.nyz were modified

\$git add file1.nyz

\$git add file2.nyz

\$git commit -m "This is commit message"

\$git commit -am "This is the commit message"

→ Add all modified files to stage then commit them.

(7) View History of repository

\$git log

```
anil0@ANIL_SOLANKI MINGW64 /d/Education/Love babber DOT WEB Dev/Projects (main)
$ git log
commit a77de50ea2796d1389881edff9621e2e5c4f7446 (HEAD -> main, origin/main, origin/HEAD)
Author: anilsolanki19 <anilsolanki192@gmail.com>
Date:   Fri Sep 12 23:04:52 2025 +0530

    HTML III all files added related to third Html Class

commit 84205d1e6b6cdb8a18e7ee02740caf42376072b2
Author: anilsolanki19 <anilsolanki192@gmail.com>
Date:   Thu Sep 11 19:38:46 2025 +0530

    HTML II class files added
```

Enter ⇒ load more
q ⇒ exit

\$git log -3

```
anil0@ANIL_SOLANKI MINGW64 /d/Education/Love babber DOT WEB Dev/Projects (main)
$ git log -3
commit a77de50ea2796d1389881edff9621e2e5c4f7446 (HEAD -> main, origin/main, origin/HEAD)
Author: anilsolanki19 <anilsolanki192@gmail.com>
Date:   Fri Sep 12 23:04:52 2025 +0530

    HTML III all files added related to third Html Class

commit ffc7e575e28787cd575784e3436f5f3d407ufd523d
Author: anilsolanki19 <anilsolanki192@gmail.com>
Date:   Thu Sep 11 19:38:46 2025 +0530

    HTML II class files added

commit fc7e575e28787cd575784e3436f5f3d407ufd523d
Author: anilsolanki19 <anilsolanki192@gmail.com>
Date:   Thu Sep 11 00:15:43 2025 +0530

    HTML - I first class's code and Resources updated
```

Latest 3 commit log

\$git log -P

```
anil0@ANIL_SOLANKI MINGW64 /d/Education/Love babber DOT WEB Dev/Projects (main)
$ git log -P
commit a77de50ea2796d1389881edff9621e2e5c4f7446 (HEAD -> main, origin/main, origin/HEAD)
Author: anilsolanki19 <anilsolanki192@gmail.com>
Date:   Fri Sep 12 23:04:52 2025 +0530

    HTML III all files added related to third Html Class

diff --git a/3-HTML/III/10_Special_characters.html b/3-HTML/III/10_Special_characters.html
index 106644..59f15f0
--- /dev/null
+++ b/3-HTML/III/10_Special_characters.html
@@ -1,1 +1,1 @@
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Special Characters</title>
    <style>
        table,
        td,
        th,
```

commits log with all modifications
q ⇒ exit

\$git log --stat

```
anil0@ANIL_SOLANKI MINGW64 /d/Education/Love babber DOT WEB Dev/Projects (main)
$ git log --stat
commit a77de50ea2796d1389881edff9621e2e5c4f7446 (HEAD -> main, origin/main, origin/HEAD)
Author: anilsolanki19 <anilsolanki192@gmail.com>
Date:   Fri Sep 12 23:04:52 2025 +0530

    HTML III all files added related to third Html Class

3-HTML/III/10_Special_characters.html | 13 ++++++-
3-HTML/III/10_Special_characters.html | 13 ++++++-
3-HTML/III/1_w_table.html | 55 ++++++=====
3-HTML/III/1_w_table.html | 55 ++++++=====
3-HTML/III/1_w_table.png | Bin 0 -> 13219 bytes
3-HTML/III/1_w_table.png | Bin 0 -> 13219 bytes
3-HTML/III/1_w_solution.html | 67 ++++++=====
3-HTML/III/1_w_solution.html | 67 ++++++=====
3-HTML/III/1_w_solution.html | 65 ++++++=====
3-HTML/III/1_w_solution.html | 65 ++++++=====
3-HTML/III/1_w_solution.html | 69 ++++++=====
3-HTML/III/1_w_solution.html | 72 ++++++=====
3-HTML/III/1_w_solution.html | 72 ++++++=====
```

each file.

Show which files were edited and How much changes happened in each file.

\$git log --one line

```
anil0@ANIL_SOLANKI MINGW64 /d/Education/Love babber DOT WEB Dev/Projects (main)
$ git log --oneline
a77de50 [HEAD -> main, origin/main, origin/HEAD] HTML III all files added related to third Html Class
84205d1 HTML II class files added
fc7c575 HTML - I first class's code and Resources updated
ed57a94 Git folders and its Notes added
2f920de 0-Git folder added and folders+Notes related to git & github shifted to this folder
53a15a0 Initial commit
```

\$git show 84205d1e6b6cdb8a18e7ee02740caf42376072b2

commit id

```
anil0@ANIL_SOLANKI MINGW64 /d/Education/Love babber DOT WEB Dev/Projects (main)
$ git show 84205d1e6b6cdb8a18e7ee02740caf42376072b2
commit: 84205d1e6b6cdb8a18e7ee02740caf42376072b2
Author: Anil Solanki <solanki.anil1995@gmail.com>
Date: Thu Sep 11 19:38:46 2020 +0530

HTML II class files added
```

(viii) Undo modifications / commits OR amend commit

\$git restore file.xyz
\$git commit -- amend

→ Remove all modifications in file.xyz
→ To edit commit message of latest commit. (I → Insert) (esc:wq+enter to exit)

\$git revert 84205d1e6b6cdb8a18e7ee02740caf42376072b2

→ Remove all modifications of a particular commit

\$git reset -- soft 84205d1e6b6cdb8a18e7ee02740caf42376072b2

\$git reset -- mixed 84205d1e6b6cdb8a18e7ee02740caf42376072b2

\$git reset -- hard 84205d1e6b6cdb8a18e7ee02740caf42376072b2

→ take the HEAD to the given commit, and delete commits present after it.

-- soft → working dir. changes show as staged

-- mixed → working dir changes show as modifications

-- hard → working dir changes get discarded. (removed)

(g) Adding files not to Track

→ Open git repo in Terminal

\$touch .gitignore → Create a .gitignore file

\$echo "hello.txt" >> .gitignore → Add text "hello.txt" in .gitignore file

```
$ git add .gitignore → add .gitignore to stage  
$ git commit -m "added hello.txt to .gitignore"  
$ touch hello.txt → Create hello.txt, & it will not  
be tracked.
```

(10) Basic Terminal Commands (Git Bash)

```
$ pwd → print working directory (current dir)  
$ ls → list items  
$ ls -al → list items including Hidden  
$ cd project → change directory to project  
$ cd .. → change dir. one level above (parent)  
$ mkdir n-folder → make dir. with name: n-folder  
$ touch file.xyz → create a file, name: file.xyz  
$ echo "Hello Dunya" >> file.xyz → put text into file  
$ cat file.xyz → Show content of file.
```

(11) Creating, Merging & deleting Branches

```
$ git branch → list all branches, with current*  
$ git branch newB1 → create a newB1 branch  
$ git checkout newB1 → switch to newB1 branch  
$ git checkout main → switch from newB1 to main  
$ git checkout -b newB2 → create & switch to newB2  
$ git checkout main → switch to main  
$ git merge newB1 → Merge newB1 branch to main  
$ git branch -d newB1 → delete newB1 branch
```

Sometime while merging branches we may get merge conflict. You can either abort merge or resolve conflict.

(a) About merge

\$git merge --abort

(b) Resolve conflict

- Let file.nyz is having conflict
- open file > resolve conflict
- Add file.nyz to stage
- Commit ~~~~~ The branch get merged.

* How to work with Branches

- switch to the branch
- develop it
- commit the changes.
- Then switch to other branch

Note:- make sure to commit changes of current branch before switching to other branch.

(12) Tagging a commit

\$git tag -a myTagName
-m "Tag message ---"

84205d1e6b6cdb8a18e7ee02740caf42376072b2

\$git tag -d myTagName

→ To add tag
→ delete the tag from the commit.

(13) pulling latest commits from Remote repo

\$ git pull → pull latest commits
\$ git pull origin newB1 → pull commits from branch newB1

Note → If we pull latest commits using \$git pull then our local changes get deleted, or may show conflict, so we use stash Area.

\$ git stash
\$ git pull
\$ git pop

→ unmerged path conflict

→ resolve conflict

\$ git add .

\$ git commit -m "pulling from remote"

(14) Pushing to Remote

- create new repo on Github.com, name: repo1
- on Local machine > Terminal

\$ mkdir repo1

\$ git init

\$ touch README.md

\$ echo "This is README.md file" >> README.md

\$ git add README.md

\$ git commit -m "Initial commit"

\$ git remote add origin https://github.com/ab17/repo1.git

```
$ git config --global user.name "ab17"  
$ git config --global user.email "ab17@gmail.com"  
$ git push -u origin main  
→ Error (Here you may be asked for username  
and Password.)  
username → ab17  
Password → Not the account password.
```

github.com > repo1 > settings > developer settings
> Personal Token Access > Token classic >
Generate new token > Generate new token (classic)
> name (xyz) > No expiry > check all permissions
 > copy token → This token is the password