

Git & GitHub

- Git is a software, which save the changes in ^{program}.

- installation :-

Step 1 → Type 'Git windows install' in browser.

Step 2 → Download from the 1st link appear as your desktop operating system i.e 64 bit or 32 bit.

Step 3 → install & launch Git bash to confirm the installation.

- Configuring Name & Email :-

Step 1 → Open Git bash terminal.

Step 2 → git config --global user.name <name>

Step 3 → git config --global user.email <email>

To check → git config --global user.name <name>

- Opening IDE :-

Step 1 → Open Git bash Terminal

Step 2 → Code . ← (code editor opened if u have installed any IDE before)

NOTES

- To get a repository whether you can use init or clone a repository.

- Initialize a repo. :-

step 1 : git init ↴ (in your git bash terminal inside a folder)
↳ ls -lapt ↴
(to see hidden files)

step 2 : git status ↴ (To check untracked files)

- Adding files to staging area ↴

step 3 : git add <index.html> ↴ (files entered into staging area)
(index.html is a file u create on that folder)

- Commit :-

step 4 : (to start a project we have to do a commit first)

git commit ↴ (A web editor opened)
(Press) I

~~e (Esc)~~ <esc> : w q

⇒ (Files successfully committed)

NOTES

- (To check untrack files)

Step 5: git status ↪ (now it shows "nothing to commit")

- Creating new file :-

Step 1: touch <filename.ext> (inside git bash)

- Adding all files in staging area :-

git add -A ↪

- Committing all files :-

git commit -m "committing message" ↪

- Recovers saved committed files :-

git checkout <filename.ext> ↪

- Recovers all files which are committed :-

git checkout -f ↪

• git log :-

(It tells about ~~the~~ author & the committed messages which have done before)

• Filtering git log files :-

git log -p - (number users want to show about committed file)

q - (for quit)

• git diff :-

(It ~~tell~~ compares working tree with staging area. If users modify a file & run git diff it shows what new text added in that particular file.)

• git diff --staged :-

(compares staging area with last commit)

• Directly add & commit :-

git commit -a -m "commit message"

- 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31
- `ls -d` (to show all files inside that folder).
 - Removing files from staging area :-
`git rm --cached "<file.ext>"`
 - Delete file permanently :-
Step 1: `git rm "<file names>"`
Step 2: `git commit -a -m "message"`
 - Short status :-
`git status -s`
 - `gitignore` :-
(Some files we don't want to track, so to ignore them, here some command :-)
Step 1 : `touch .gitignore`
Step 2 : (The files we want to ignore, we have to type the file name.ext inside git ignore folder)
Step 3 : `git commit -a -m "ignores and added <gitignore> message"`

Some commands inside .gitignore :-

- mylog.log \Rightarrow (it ignore every file that named like that)
- /mylog.log \Rightarrow (it only ignores that file whose git ignore folder exists, it never ignore such named file from any folder)
- *.log (ignore all which has extension .log)
- ignore/ (it ignores that folder, which named 'ignore') & its file

o Branch :-

Step 1:- git branch feature1 \leftarrow (it creates a new branch 'feature1')

Step 2:- git branch \leftarrow (to check the branches)

Step 3:- git checkout feature1 \leftarrow (to switch the branch from 'master')

Step 4:- (now user can do any change but no effect on master branch) branch

~~step 5 :- git checkout master~~ ↵ (to switch
on the
master branch)

~~step 6 :- git merge 'feature1'~~ ↵
(if user wants to merge
both branch, this command can be
used)

• `git checkout -b <new branch name>` ↵

(this command create new branch
as well as switched to that branch also)

2014

DECEMBER
FRIDAY

WK 52-360-005

26

SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7	8	9	10	11	12	13	
14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31										

Git Hub

- Type 'GitHub' on browser
- Create account
- Create repository by clicking '+' sign at the top right in GitHub home page.
- Copy "git remotegit" text which appears after creating repository.
- Paste on the git bash terminal ↪

git bash:-

(After pasting) →

Step 1: git remote ↪ (It shows origin because origin is short name of that

Step 2: git push origin ^{master} ↪ (it shows 'no repository found' if this is new sign in in GitHub)

JANUARY 2015

WK 52-359-006

2014

SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT
11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31							

25

DECEMBER
THURSDAY

o How to get read write access in GitHub?

Step 1 :- go to settings

Step 2 :- go to 'SSH & GPG Keys'

Step 3 :- click on 'generating ssh key'

Step 4 :- go to 'generating ssh key--"

now & copy "2nd point accordingly.

Step 5 :- paste it on browser.

Step 6 :- edit the link with your email
~~in git~~ which you given while config. email in git bash terminal.

Step 7 :- now copy that url & paste on the git bash terminal. ↪
↪

Deploying that generated key :-

Step 1 :- go to 'eval \$..' & copy it.

(You find it on that ssh key page
scroll down)

Step 2 :- paste in git bash ↪

(U get ur process id)



Step 3 :- copy "ssh add - -" & paste
(find below eval in that page)
again on Git bash

Step 4 :- press enter

Step 5 :- Go to "Adding a new SSH Key"
in Git Hub page.

Step 6 :- copy the clip that appears
1st & paste it on Git bash (don't press ↵)

Step 7 :- now your command : CAT < paste
file > ↵

Step 8 :- copy that key.

Step 9 :- paste on SSH key page →
New SSH key in
Key text area.

*changing URL :-

Step 1 :- Go to SSH in the home page
of Git Hub.



Step 2 :- copy that url.

Step 3: (In git bash, type →)

git remote set-
-url origin <path>

Step 4: git push -u origin master

<Files successfully pushed>