

Record the things or list
you might want to do.



IMPORTANT

NOTES

Git

R

GITHUB

Din Ma Har Din Ek Git tutorial and

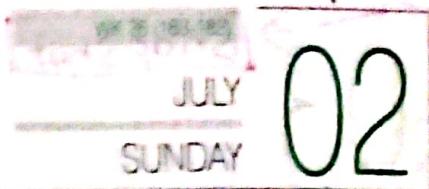
Javascript tutorial our 3 ke
baad MICTET ki Tutorial

(

7/ May/ 22

AUGUST - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
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								



- ① All Question & Doubt are write in :- Blue Pen
- ② Ans & Solution write in :- Blue Pen
- ③ And my Suggestion & opinion & tip & trick
Write in :- Red Pen.

* What is a Git ~~VCS~~ Version Control System.

→ Git is Version Control Software System.

- Easily recover a file
- Who introduced an issue and when?
- easily Roll back to previously working state!

* THREE Type of ~~VCS~~ (version control system).

⇒ Local VCS

⇒ Centralized VCS

⇒ Distributed VCS.

Git is

2017

• git => Me ap ki all project ki history hoti hai

Hidden folder 03 Snapshot JULY MONDAY

JUNE - 2017											
M	T	W	T	F	S	S	M	T	W	T	F
			1	2	3	4	5	6	7	8	9
12	13	14	15	16	17	18	19	20	21	22	23
24	25	26	27	28	29	30					

In Git

* **Remote** => ~~internet, Server, Repository~~

* **Checksum** => Gap ki original file ko gap me server par push ki ha to file ka Checksum hota hai ~~unique~~ ^{digit} then you pull a file in local then the checksum is not match that mean the file is edit by some one, and the checksum is same that means file is not change file is original.

X-X-X

First Time Use Git
you tell who are you, in Git

① for Name.

⇒ git config --global user.name "Ashish"

② for Gmail. Id.

⇒ git config --global user.email "ABCxyz@com"

* In Git You can use different
Editor

⇒ git config --global core.editor vim

↳ ALSO you can use Linux in
git

X-X-X

05

JULY
WEDNESDAY

JUNE - 2017

M	T	W	T	F	S	S
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		

Three Stage Architecture ✎

① Working directory → jsapar aap work karane ho.

② Staging area → ~~the~~ files hoti ha jo aap change ho ki next Commit kyu chayega.
 ↳ (Snapshot)

③ git directory → ~~apko~~ जाप ने window par Ek hidden file hoti hai .git ⇒ git ka kaam

Ha ki aapki sare project ka snapshot Rohete ha agar aapko ~~previous~~ previous file ko Roll Back karsakte hai

2017

AUGUST - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
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								

WK 27 (187-195)

JULY

THURSDAY

06

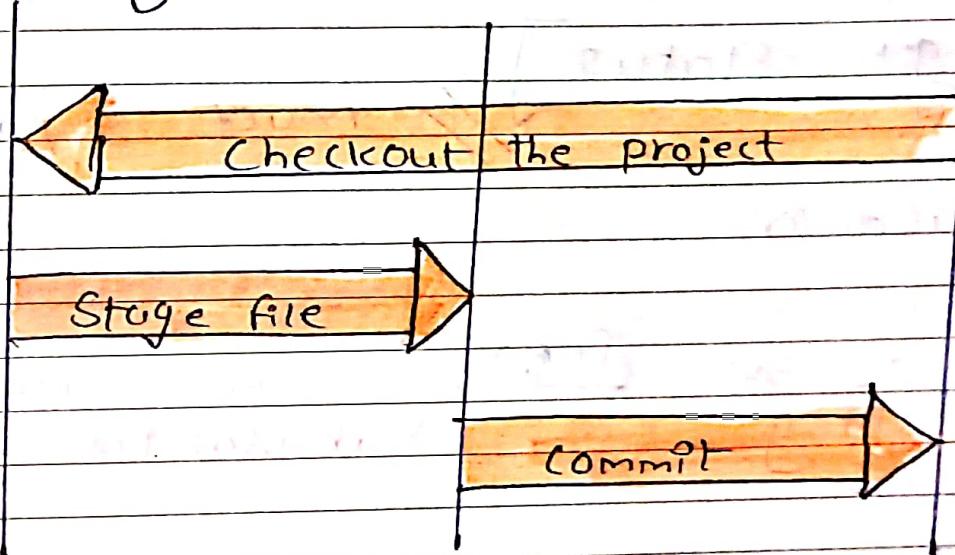
Local Operations

8

9 (Working
directory)

(Staging
area)

(git directory
(repository))



2017

9/5/22

JUNE - 2017

07

[DRB-177] WK-27

JULY

FRIDAY

M	T	W	T	F	S	S	M	T	W	T	E	S	S
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									

* Basic Command of Git *

=) In git always Run.

git status

// must important

↳ Rationale

* Saap koi bhi ~~folder~~ ^{repository} ko git
repository nahi bana saktा ho.

=) You first ~~can~~ write command.

=) **git init**

→ When you initialized git init then you have created a hidden folder **.git**

// Then your folder is initialized for repository // tracking
by Stash

AUGUST - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S	S
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										

WK 27 (89/176)

JULY
SATURDAY

08

* What is a branch master.

=> Branch master, (main branch)

* git add --a / git add .

(i) git add --a command is add all files in fold folder in staging or git

* git commit -m " " "

(i) git commit -m " " is a next stage level ~~to~~ and It will take a snapshot of you file in git folder.

& also compulsory to write commit in semi colon, for you understand what you are changing

2017

11/5/22

AUGUST - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S	S
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										

WB 2B (19-174)

JULY

MONDAY

10

* Cloning a Remote Git Repository From Github *

=> **rm -rf .git**, (use ~~soft~~ soft)

(↳ deleted all folder contain of repository & .git file also.)

* **pwd** (present working directory)

(↳ is a Basic command.)

* **ls** (list content)

(↳ list of all the files & folders in the current directory)

* **cd** (change directory)

(↳ change directory to other.)

11

第17章 故事

12/5/22

JULY

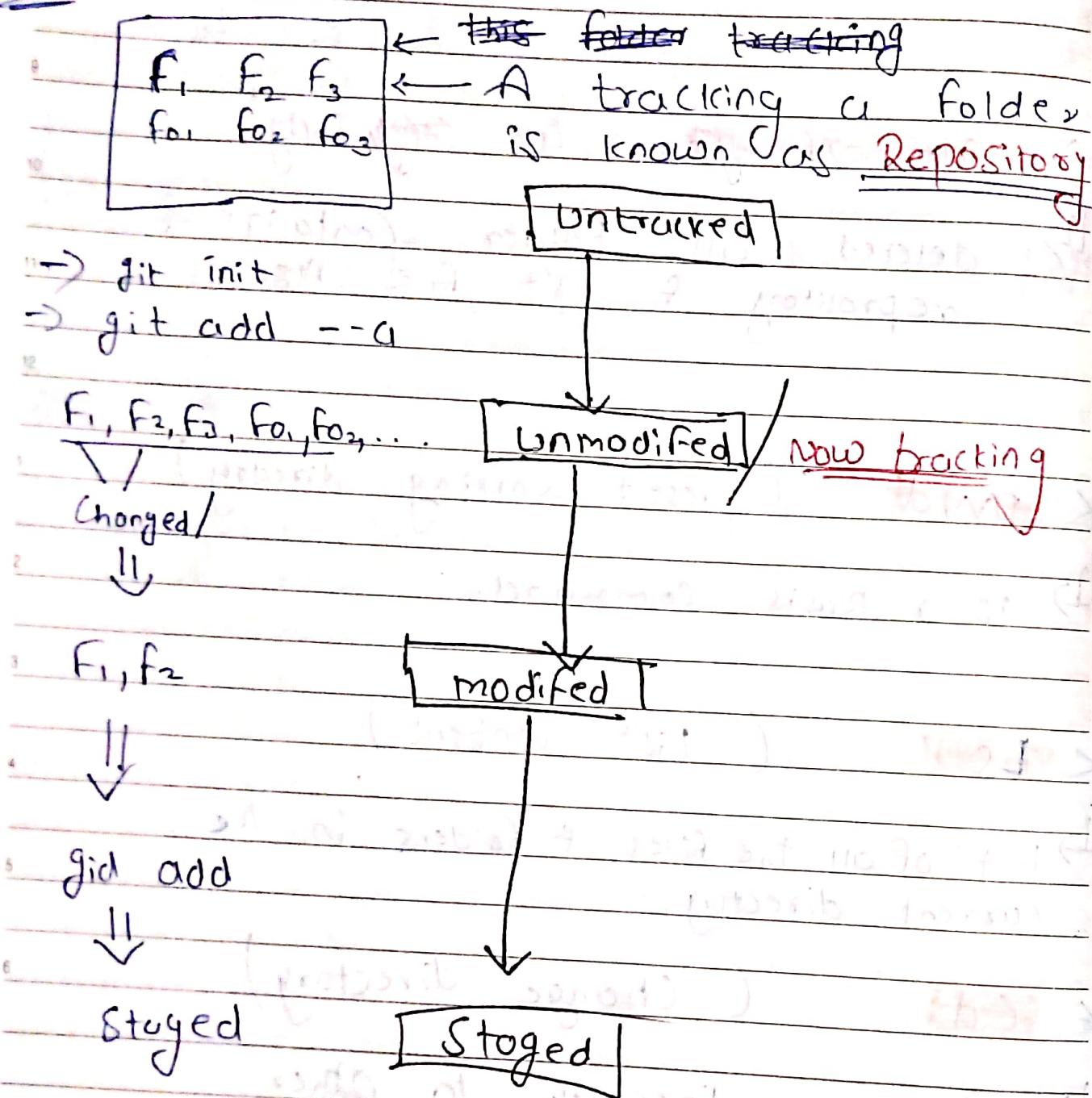
TUESDAY

JUNE - 2017

JUNE - 2017												
M	T	W	T	F	S	S	M	T	W	T	F	S
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							

Files Status & Lifecycle

688



2017

AUGUST - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
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								

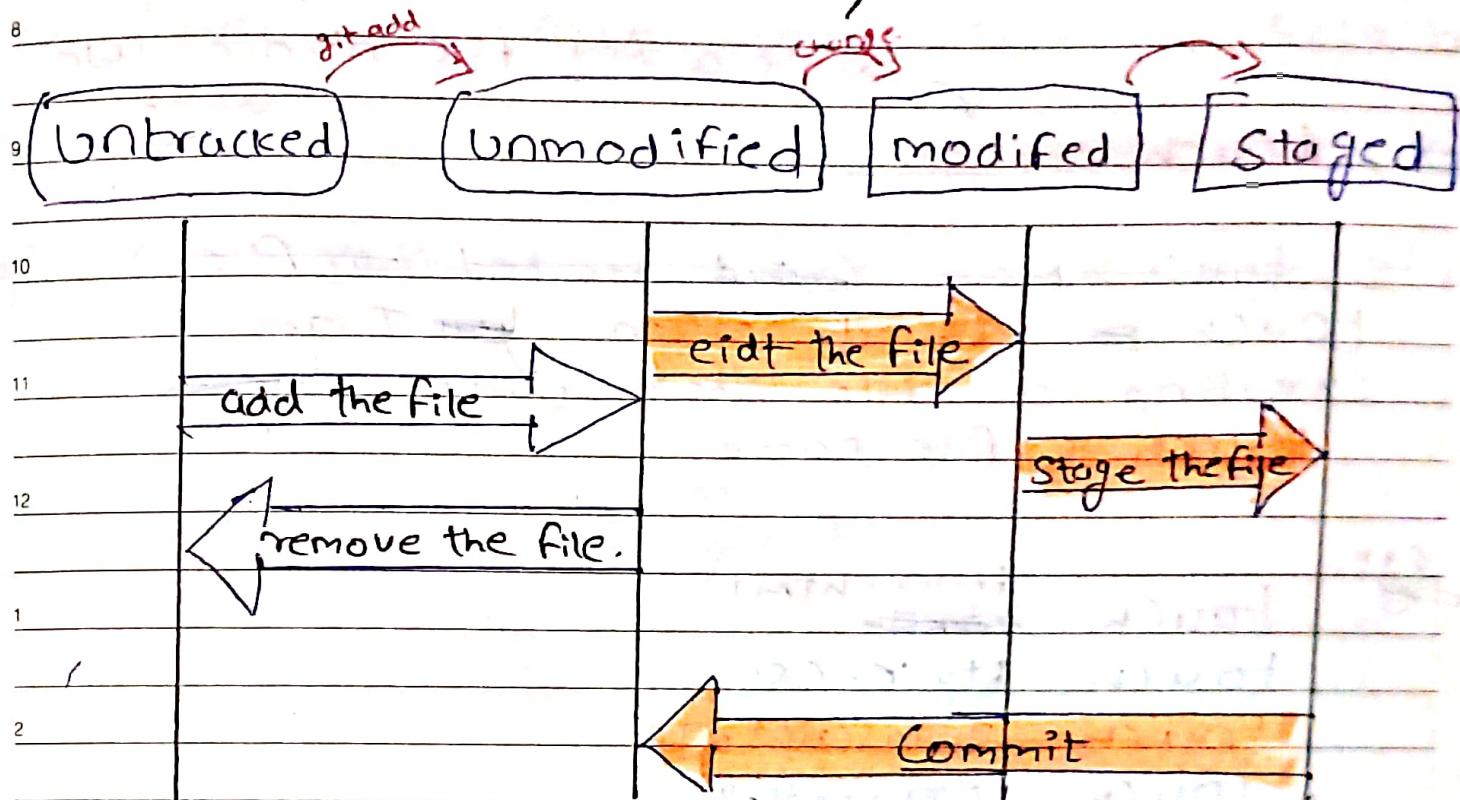
WEEK 28 (80-172)

JULY

WEDNESDAY

12

File Status Lifecycle.



The Git Workflow

Creating, testing, and releasing software is not just about writing code well. It's also about creating a healthy workflow that can support the development process. This is where Git comes in.

Git allows us to manage our code in a very efficient way. It provides a way to track changes, collaborate with others, and release software quickly and easily.

2017

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S	S
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									

13

JULY
THURSDAY

git ignore files.



write a Any file name we want.

* touch

↳ touch mean, bread created a file
touch command mean, I am
creating a file then write touch
touch & file name

Eg:-

touch index.html

touch style.css

touch Javascript.js

touch error.log

* why .gitignore file created.

⇒ mablooo mera pass tas punda ka
files ho us mein se kuch file ko
commit nahi karna hai to
main use .gitignore file main
dal dunga.

⇒ uske liya .gitignore file ko
created kard padega in repository.

⇒ Aur then write a name which
file you want to ignore in repository.

AUGUST - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
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
28	29	30	31									

WEEK 28 (WEEK 17)

JULY
FRIDAY

14

Write in .gitignore inside a file.

then the are ignore in my Repository

Also ignore multiple file & folder in .gitignore

*.log file & folder

*.txt file & folder

(All) use tarha ke sam file & folder ko ignore kro

13/5/22

15

JULY

SATURDAY

JUNE - 2017											
M	T	W	T	F	S	S	G	M	T	W	F
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					

* git diff = Showing changes between commits/staging Area & working Directory

* git diff :- git diff compared to working directory to staging area.

↳ Agar main git add command run kiyा our main board, suppose main EK .txt file ko modify kiyा our main board. git status run kiyा too, (mujhe tu commit dikhaya ga EK staging area ka our EK modify ka.)

↳ our main maane git diff Run kiyा to previous file .txt our modify file .txt main different dikhaya kiyा.

* git diff --staged :- will only show changes to files in the "staged" area.

AUGUST - 2017

19/5/22

M	T	W	T	F	S	S	M	T	W	T	F	S	S
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										

JULY
SUNDAY

16

* Git : Skipping The Staging Area *

* git commit -a -m " " "

(b) This command only apply on tracked files and directed to staged, untracked files is not go in commit

⇒ Staging Area koi commit bolata hai.

17

(100-107) WED

JULY

MONDAY

15/5/22

JUNE - 2017

M	T	W	F	S	S	M	T	W	F	S	S
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					

* Moving and Renaming Files in Git

Remove
* `git rm nameoffile`

↳ `git rm` is a command is Remove the file in repo and also add in stageding staging Area.

- eg:- ① `git rm first.txt` \Rightarrow Remove file
- ② `git rm index.html` \Rightarrow Remove file.

* `git mv oldfile newfile`. \Rightarrow Rename command.

↳ `git mv` command is Rename the files and also add in Staging Area.

Eg:-

① `git mv first.txt Second.txt`

② `git mv index.html style.css`.

AUGUST - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
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								

MONDAY

JULY

TUESDAY

18

* git rm --cached name of file.

↳ git rm --cached command it is
untracked the file in git repository.
To remove a file from a git repository
but not remove in origin folder.

Eg:-

- ① git rm --cached second.txt
- ② git rm --cached style.css

also use git commit -m "rm file"

git add .

3

4

5

6

2017

19

TUE JULY 18 2017

JULY

WEDNESDAY

17/5/21

JUNE - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
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							



Git Log : Viewing & Changing Commit

* git log -p :-

↳ Show Commit who are
Change Author name & time.

* git log -p -10 -

↳ Show all Commit what
Change Author name & time and also
what file include and deleted.

* git log --stat

↳ Show all change in Short in Summary

M	T	W	T	F	S	S	M	T	W	T	F	S
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								

JULY

THURSDAY

20

* git log --pretty=oneline :-

↳ naam se jata jata hai ki;
All commit show in oneline

* git log --pretty=short :-

↳ Some All Commit Show in short way
Author name & ~~date~~ & message show

* git log --pretty=full :-

↳ Some ~~All~~ Commit Show

Some Show all Commit Similar to
Short But Extra Author & Commit &
message.

* git log --since=2 day ↳ Change which day
you want

↳ git log --since show day wise
Commit, day, week, ~~month~~, years, ~~so~~ I can also
write

2017



JUNE - 2017

M	T	W	T	F	S
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		

21

JULY
FRIDAY

Screen in googol

Please

* git commit --pretty=format:

* git commit --amend :-

4)



Enter :

Enter your Commit

Enter :

① ESC

② :wq

Open hog9
Editor.

AUGUST - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
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								

JULY
SATURDAY

22

~~Unstaging of files in Git~~ Unmodifying Git

* git reset -- staged filename
↳ to unstage a modify file

* git Checkout -- filename
↳ git Checkout -- Command is Best
Because it is Roll Back previous file, in case you deleted working stage file.

↳ git Checkout -f
↳ which file you deleted present and roll back previous file

* git checkout -f

↳

↳

SSH key GitHub

23

JULY
SUNDAY

M	T	W	T	F	S	S
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					

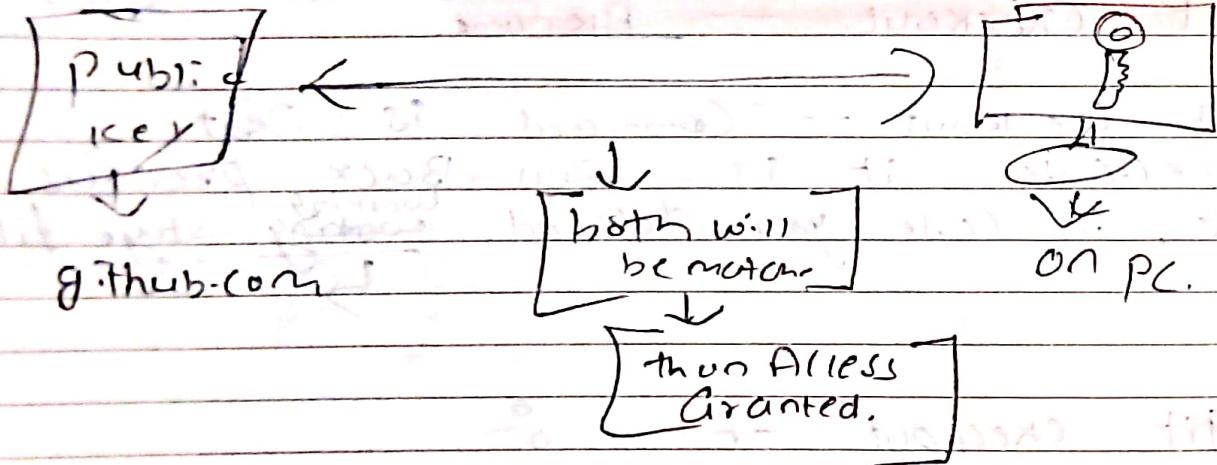
* GitHub : Working with Remote Repositories *

* Git remote -

~~SSH key generation~~

①

private key



- ① Generating a new SSH key
- ② Adding the ~~SSH~~ key to your GitHub account.

AUGUST - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S	S
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										

MR 30000-100

JULY

MONDAY

24

Q) What is a SSH?

Ans) Secure Shell → to create a secure channel between a local pc & Remote PC.

2017

25

22 | 05 / 22

JULY

TUESDAY

JUNE - 2017											
M	T	W	T	F	S	S	M	T	W	T	F
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					

Setting Alias

* Alias :- me upane man & function ko Bunasaktha hu.

Command

write which
you want a
short name

* git config --global alias. gitstatus

Write which
command you
changing.

22/08/22

AUGUST - 2017

M	T	W	T	F	S	S	M	T	W	T	F	S
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								

MONDAY

JULY

WEDNESDAY

26

Creating & Switching Branches In Git

8

* How to change Branch / Switching

9

git checkout -b develop
↓
New Branch → Branch name.

10

anything you
branch name.

* git checkout master ↳ ↳ Switching

* git checkout develop Between

11

* git branch -s - See how many
Branch you created.

12

write your Branch when
you merge.

13

git merge develop

2017