

फरवरी	
सोम.	मंगल.
मंग.	बुध.
गुरु.	शुक्र.
शनि.	
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

HARE KRISHNA!

जनवरी 2019

मंगलवार

1

→ GIT Commands

- ① git init
- ② git status
- ③ git commit -m "message"
- ④ git add . or git add --a
- ⑤ git rm -rf .git

STAGING Area:- Area , which is ready for commit.

GIT FILE status lifecycle.

बुधवार

2

UNTRACKED

UNMODIFIED

MODIFIED

STAGED

→ creating .gitignore file.

and if I add th file name inside th .gitignore file then , it will ignore that log file.

To create .gitignore file

paste this command

touch .gitignore

जनवरी 2019

३

गुरुवार

2018

तिथि	सोम	मंगल	बुध	बृह	शुक्र	शनि
30	31					
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

* one more case, If I have more than one log files then, write *.log inside the .gitignore file, then it will ignore all the log files..

→ so, If I want another file, or directory just, give the name inside the .gitignore file.

* by Default, the git ignores all the blank folders, (If no content is present). It will not show in tracked region.

४

शुक्रवार

* If I want to ignore all the folders of a particular name, even if it is inside in a directory then, we use

a ~~the~~ formated name inside the gitignore which is nospace
folder/file name /

→ / → slash means any file name with this will ignore. This slash is imp

If I want to ignore the file which is outside but do not want to ignore inside any directory file,

2019

फरवरी

रवि.	सोम.	मंगल.	बुध.	गुरु.	शुक्र.	शनि.
	1	2	3	4	5	6
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

जनवरी 2019

शनिवार

5

then use

[filename]

* git diff → This command gives me the difference between the working directory and the staging area.
command :- [git diff]

-) To look the difference (comparison) of previously staged commit and with the current staging area.

command is :- [git diff --staged]

रविवार

6

To directly commit from tracked or staging area to commit.

use :- [git commit -a -m "Direct commit"
↓
message.]

* git log :- gives all commits done, up till now.

जनवरी 2019

7

सोमवार

2018

दिसंबर

सोम.	मंगल.	गुरु.	वृश्च.	शुक्र.	गणेश.
30	31				1
1	2	3	4	5	6
7	8	9	10	11	12
14	15	16	17	18	19
21	22	23	24	25	26
28	29				

☞ To rename any file

git mv filename newfile-name

e.g:- [git mv First.txt first-renamed.txt]

To not checked (or) to not track any file if we are modifying , even if it inside .gitignore.

then , it will show the modification in git status.

8

मंगलवार

To make file from tracked to untracked then,

use :- git rm --cached file-name

e.g:- [git rm --cached db.accdb]

☞ To remove git repositories .git

git [rm -rf .git]

		1	2
3	4	5	6
10	11	12	13
17	18	19	20
24	25	26	27

जनवारा 2019

बुधवार

9

* To clone a repository.

git clone repository-link

• to make another folder
then just give same command.
and write file/folder name, when
this repository is cloned.

→ git log gives all commits up till now.

git log -p → gives all commits with
git diff.

git log -p numbers

गुरुवार

10

↳ e.g. :- **git log -p -2**

git log --stat

↳ give some short summary
of commit.

* **git log --pretty = one line**

↳ gives all summary in one line.

2018

नंबर	सोम	मंगल	बुध	बृह	शुक्र	शनि	पूर्णि
30	31						
2	3	4	5	6	7		
9	10	11	12	13	14		
16	17	18	19	20	21		
23	24	25	26	27	28		

जनवरी 2019

11

शुक्रवार

Author means who created that file.
commit → who have changed that file

* We can add some filters in git log.

git log --since=2.days



This command, will give summary of commits of previous two days.

We can write 2.days, 2.weeks, 2.months
any numbers with these.

12

शनिवार

* git log --pretty=format:"%h - %ae"



These are some types of placeholders go, on git documentation, which is inside pretty.



git commit --amend

↳ This is used to amend the changes

2019

फरवरी

गुरु	शुक्र	मंगल	बुध	गुरु	शुक्र	शनि
1	2					
8						
15						
22						
29						

जनवरी 2019

रविवार

13

1 To come outside the new terminal press **esc** then type **! colon**, then type **wq** then enter.

* To unstage file from staged file, then:

git restore --staged file-name

* To restore, the previous commit of any file, if you have by mistake committed, then,

you can use:-

सोमवार 14

**git checkout --
 ↓
 space
 ↑**

This, is used for only particular file

git remote :- This is a command in which

remote, is concerned with the remote repository. It is a shared repository that all team members use to exchange their changes.

2018						
सूर्य चतुर्दशी शुक्र शनि शनि शनि						दिवं
30	31	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	2

जनवरी 2019

15 मंगलवार

SSH Keys GitHub

You can access and write data in repositories on GitHub.com using SSH (secure shell protocol). When you connect via SSH, you authenticate using a private key file on your local machine.

To remove any file

`rm file-name`

16 बुधवार

To remove any folder, which is not a git repository.

then use :-

`rm -r foldername`

If that directory/folder is a git repository that you have pulled or cloned from the GitHub or any platform.

फरवरी									
२०१९									
१	२	३	४	५	६	७	८	९	
१०	११	१२	१३	१४	१५	१६			
१७	१८	१९	२०	२१	२२	२३			
२४	२५	२६	२७	२८					

जनवरी 2019

गुरुवार

17

So, write this command.

First ①

`rm -rf folder-name/.git`

second ②

`rm -r folder-name`

These, are the two steps.

→ To learn, ~~with~~ command type.

`ssh keys github`

★ GIT Alias

→ We can make big or lengthy commands into short forms using alias.

शुक्रवार

18

For example:- `git status`
to
`git st.`

so, To make alias, the command is.

`git config --global alias.new-form original command.`

so eg:- `git config --global alias.st status -s`

alias → not alias

जनवरी 2019

19 शनिवार

So, if i write 'git st', then, it will give the same functionality of git status.

To remove any alias, then edit.

`git config --global --edit`



This will open new terminal, just move the cursor to alias, by button up and bottom, left and right, and then just use Del button to delete.

20 रविवार

Then esc after deleting.
for alias, press esc, q, then Wq for exit.

To view all git alias:-

`git config --global --get-regexp alias.`

2019						
मार्च						कारवारी
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				

जनवारी 2019

सोमवार

21

Extra knowledge :-

vi :- visual Editor

vim :- enhanced version of vi.

These both are popular text editors commonly used in unix-like operating system including linux

Exit commands

For vi and vim :-

→ press 'ESC' key

→ Then, ':wq' to save and exit

मंगलवार 22

AND :- ':q!' to exit without saving any changes

For nano :- eg:- bashrc

→ Ctrl+X

For visual studio code

→ press 'Ctrl+S'

Ctrl+H or Ctrl+F4 to
Close the
file tab.

2018

रवि	सोम	मंगल	बु.	गु.	पू.	शु.
30	31					
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

जनवरी 2019

23 बुधवार

→ To exit the VS Code application
press Ctrl+Q

* For Notepad:-

→ press 'Ctrl+S' to save your file.

→ press 'Alt+F4' to close the
Notepad window.

* For Linux Terminals,

→ 'Ctrl+X', 'Alt+C' or 'Ctrl+D'

24 गुरुवार

other than git alias command we
can create our own custom script.

or functions in shell configuration
file:- '.bashrc', '.zshrc')

① To open shell configuration file
using txt editor, for bash it's is

'.bashrc'

2019

फरवरी

वर्ष	सोम	मंगल	बुध	गुरु	शुक्र	शनि
३	४	५	६	७	८	९
१०	११	१२	१३	१४	१५	१६
१७	१८	१९	२०	२१	२२	२३
२४	२५	२६	२७	२८		

जनवरी 2019

शुक्रवार

25

command :- nano ~l.bashrc

② Add the following function

eg:- `gcl() {
git clone "$1"
}`

③ To apply the changes and exit,

press Ctrl + X, then just press

Enter.

शनिवार

26

④ Run again this command

source ~l.bashrc

Now, we can use our own alias ;

eg:- `gcl <repository-links>`

जनवरी 2019

27 रविवार

"\$1" → Indicates that the function will take one argument.

To create a new branch

`git checkout -b new_branch_name`

eg:- `git checkout -b develop`

To switch branches, just use

`git checkout branch-name`

28 सोमवार

eg:- `git checkout master`

→ To show all branches.

Type command :-

`git branches`

* → This indicates the current branch.

२०१९

फरवरी

१	२
३	४
५	६
७	८
९	१०
११	१२
१३	१४
१५	१६
१७	१८
१९	२०
२१	२२
२३	२४
२५	२६
२७	२८
२९	३०

जनवरी २०१९

मंगलवार

२९

Whenever we make branches other than master branches, then make sure any changes in that branch should be committed.

→ How to merge the branches in master branch.

command:-

`git merge branch-name`

e.g:- `git merge mycleanup`

बुधवार

३०

Merge conflicts is when, we do changes in master branch, and that component changes is also there in another branch, so it tells user that which changes you want.

"current change", "Incoming change" or "both change"

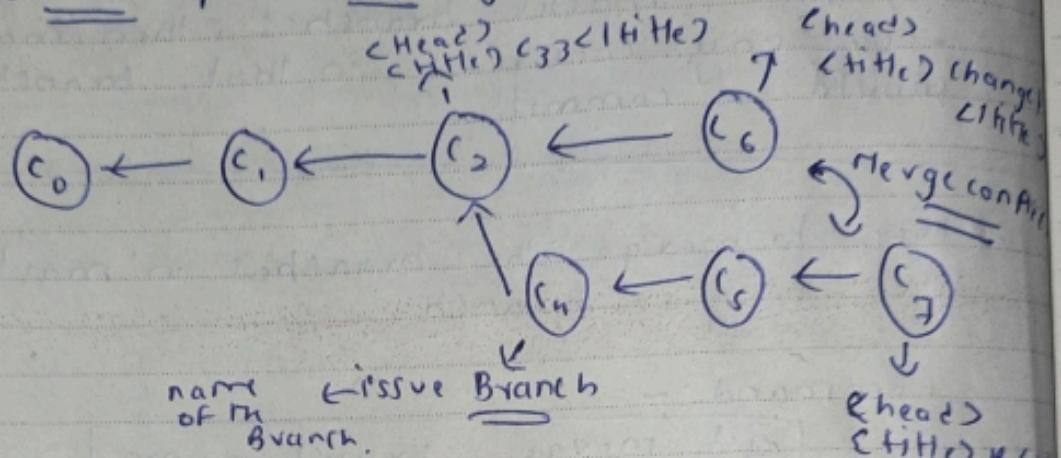
2018

सं.	१०	२०	३०	४०	५०	६०	७०	८०	९०	१००	११०	१२०	१३०	१४०	१५०	१६०	१७०	१८०	१९०	२००
30	31																			
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5

जनवरी 2019

31 गुरुवार

* Best example for merge conflict.



* conflict Resolution markers

<<<< head: index.html

=====

>>> issue1: index.html

* Branch Management

If you want to see the last message (commit) message of branch, then

use this command:-

git branch -v
commit hash

commit message

विकास
2019

मार्च

१	२
३	४
५	६
७	८
९	१०
११	१२
१३	१४
१५	१६
१७	१८
१९	२०
२१	२२
२३	२४
२५	२६
२७	२८
२९	३०

फरवरी 2019

शुक्रवार

1

git branch --merged → already merged branches.

git branch --no-merged → not merged branches.

* Deleting branches.

git branch -d branch-name → give error if develop branch is not merged.
 ↓
 So, To avoid error use -D → Capital D.

शनिवार

2

BRANCHING WORKFLOW

4

long Running Branches

- ⇒ master
- ⇒ Develop
- ⇒ pu

Topic Branches.

→ Replace Text
 With Type It.
 ⇒ Typejs Int.

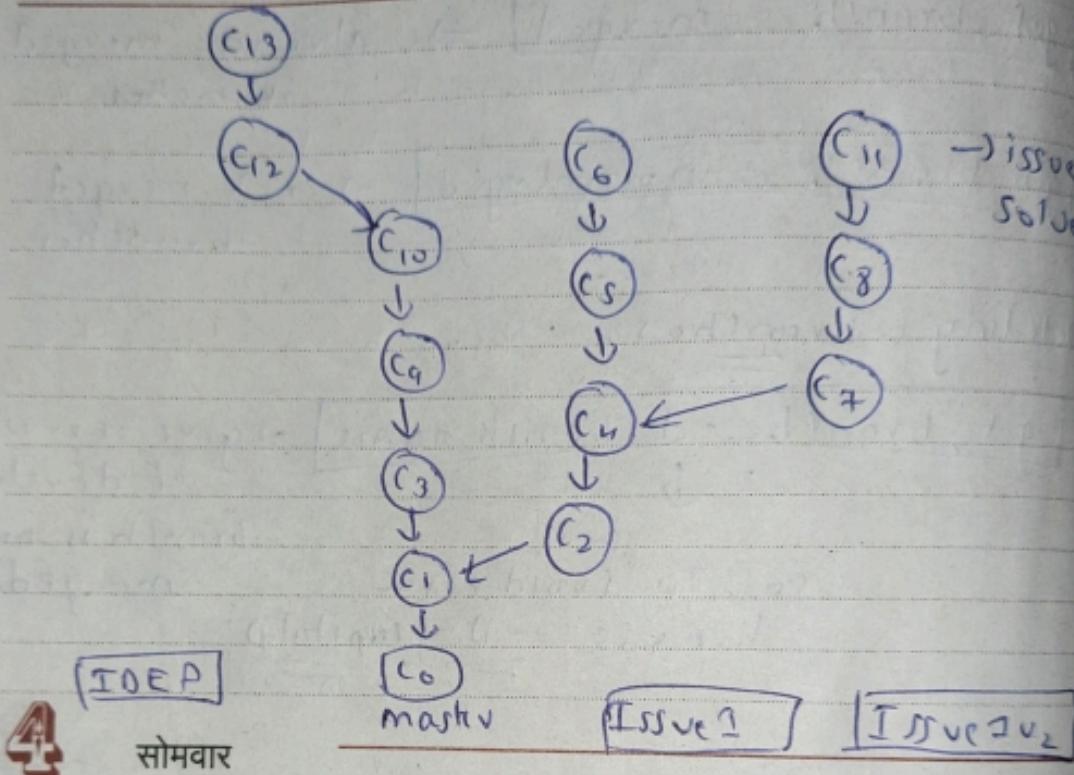
2019

तिथि	गोम	मंगल	बुध	बृह	शुक्र	शनि
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		

फरवरी 2019



रविवार



☞ When we push the code / repositories in github, then in github when we check the branches, then there is only one branch which is master branch, and if you want to have it then you need to explicitly do this for publishing.

so, you need explicitly push
e.g. git push origin branch_name

2019		मार्च	
३	४	१	२
०	१	३	४
७	८	५	६
४	५	१२	१३
२५	२६	१९	२०
१	२	२८	२९
३१		३०	
रवि	सोम	मंगल	बुध
पूर्ण	गुरु	शुक्र	शनि

फरवरी 2019

मंगलवार

5

→ Make sure you should be in that particular branch for pushing that code.

To rename any different name in Branches of remote then,

git push origin bugfix: mybugfix

To delete the branches from

remote repositories

बुधवार

6

then:-

git push -d origin bugfix

git push -d origin mybugfix.