**Git/GitHub**

-----------------------------------------------------------------------

git --version

**-Check user name and email**

git config --global user.name

git config --global user.email

git config --global --edit edit username and password

To exit from vim

Esc

:w save

:q exit

**>If want to make any changes in name and email**

git config --global user.name "new user name"

git config --global user.email "new user email"

**>Create folder for git repository**

mkdir new folder

cd new folder

git init :git initialised

ls

ls -a : .get repository is created

**>Create new file in repository Hello.java**

git status :will tell what changes made in this repository

**>ADD**  File Hello.java is created but not added yet

git add Hello.java :add file which is not added in git

git add --all :Add multiple file Note: The shorthand command for git add --all is git add -A

git status :files are added to the Staging Environment, and we are ready to do our first commit.

new file Hello.java

**>COMMIT**

git commit -m "initial commit should enter meaningful comment"

Git Commit Log:-To view the history of commits for a repository, you can use the log command:

git log :how many commit has made so far hashcode is generated for each committed

**Now make some changes in file Hello.java or project**

git status

modified Hello.java

**Now create one file Driver.java**

git status

modified: Hello.java

Untracked files: Driver.java

**Now add changes**

git add Driver.java

git status

still Hello.java is not in staging because not added yet so at this stage if we commit any changes, than only one file will be committed that is Driver.java

**Now commit new file with comment**

git commit -m "Driver java class created"

Now two commit hashcode is visible commit 98a959b70c72320edfdc196ac9df9dbfc1c0b13b (HEAD -> master) here HEAD move commit to commit basis

[

git help --all See All Possible Commands

cmd + k /clear cls

Use fish shell extension for auto code

]

git add . :all file staging me aa jayegi

git log :will show all commit so far we have make 3 changes

**>Now we can go to at any changes time in back date&time**

If we want to go at initial commit than we have below commands

Currently showing hash code is latest commit so got to below one

**>Go to previous change:**

git checkout <commit hash code/branch name> :previous date/time code can checkout. Branch name example "commit e95f195769dd391ec9d8d2dff0a51c48139f8393"

git checkout e95f195769dd391ec9d8d2dff0a51c48139f8393"

After checkout we are in initial commit and the current changes is not in initial commit

By using checkout we can go at any state of changes made

Now check branches

D:\git\_tutorial\_vscode\_demo> git branch

\* (HEAD detached at e95f195)

master

git log :it will show only current stage changes. if one commit then show only one commit

**>Now go back to master/current stage**

git checkout master :come back to current face

git log :now again all commit is visible to us

**>How to create a new branch?**

Basically there is master branch from this we can create multiple branches

:We can create many branches from the master branch, So that every developer can work on their own branch.

So that changes made in one branch will not impact on others.We can also make many branche from one branch.

In production there is one 'master' branch and one is 'dev' branch

**>Create branches**

git branch <branch name>

:git branch dev :new branch created

git checkout dev :now we are in dev also not recommended to work here so create one more branch.

git checkout -b <branch name> :new branch will create and directly checkout

git checkout -b rajkashy/multiply :so we can recognise who is working on what/So whatever changes we will make here will not impact to other branch

git status

git add newfile.java

git commit -m "added newfile.java"

git checkout dev :we are now in dev branch/

git log :changes made in rajkashy/multiply will not display in dev.

git merge rajkashy/multiply :now this branch will merge in dev branch

git log :all changes will now appear in dev also

**Note:->**Before commiting in the master branch first confirm in dev that everything is working only then make changes in master branch.

git checkout master :now in master branch

git log :master didn't know changes made so far

Now confirm everything is working fine than merge in dev only

git merge dev :dev branch is merge in master

git log :now all changes from dev is in dev now than we send master to production

Basically multiple programer can work on different branch and each branch is independent from each other

**>IGNORE**

Ignore file :If we do not want to send it to git ecosystem

secretkey.txt :create file which we want to ignore

touch .gitignore :now file .gitignore is created and can put secretkey.txt file here

or can directly create file .gitignore

git status

.gitignore

secretkey.txt

now write file name(secretkey.txt) in ".gitignore" file will not display in git status and git will not track that file

we can also put .gitignore in .gitignore so this file also will not track

git status

--------------

**>GitHub :now come to the github in you profile**

-------------

**Add local repository to github**

-create one branch can create same name which is declare in local repository after creating github will ask what we want to do

also mention command for local repository and new repository follow commands

Create new repository with same name as git -->describe->public/private etc create repository

now it will ask for setup:-what we want to do-> where we want to create a new one or a repository that already exists.

…or create a new repository on the command line/…or push an existing repository from the command line/…or import code from another repository

echo "# git\_tutorial\_vscode\_demo" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/rajkashy889/git\_tutorial\_vscode\_demo.git

git push -u origin main

So in case we have already repository created so we will follow below steps

…or push an existing repository from the command line

git remote -v :initially it will not display anything

git remote add origin https://github.com/rajkashy889/git\_tutorial\_vscode\_demo.git

git remote -v :now it will tell the origin for fetch or push

origin https://github.com/rajkashy889/git\_tutorial\_vscode\_demo.git (fetch)

origin https://github.com/rajkashy889/git\_tutorial\_vscode\_demo.git (push)

now origin has been created where we can do push or pull

git branch -M main (master):push

git push -u origin main (master):changes push in origin(origin name could be anything but its convention)

will ask for authentication enter your credential it ask for first time

CONGRATULATIONS YOUR FIRST COMMIT HAS BE DONE

Repeated code

[Copy commands and run from vscode terminal or cmd FOR MASTER

git remote -v

git remote add origin(can give any name but its convention) https:...

git remote -v

git branch -M master

git push -u origin master

-will give error if try to push with different user name so first remove from keychain and login with current user and password

enter github username:

and password

Fist time will ask user and password but later it will store in key chain

Congrats first commit is done.]

**Now make any changes**

git add filename.java

git commit -m "msg"

git push (here no need to write because by default its origin and master)

**Now we have multiple branch to push, do same thing**

git checkout dev

git push -u origin dev (dev branch will also push )

This time it will not ask credentials

now got to github dev branch is also pushed

**Do same for rajkashy/multiply branch**

git checkout rajkashy/multiply

git push -u origin rajkashy/multiply

**How can multiple developers work in a project?**

There is two way

1.make is open source, so developer can fork and make changes and send pull requests

2.or if we want developer can make changes in same code for this we can add collaborator

Setting->manage access->invite a collaborator>enter name/email

now developer can access same repo and clone on their own repository and make changes ->than changes can push

**How can we contribute to others' code or fork ,(clone, pull, push)?**

**Add others code to our repository**

Create a new repository public for open source anyone can access this repo->can fork any one and contribute.

Now those who want to contribute he can fork so new repository will created on his github->now can make any changes in it->

**Now clone on our local repo**

copy https: link from projet

come to git vscode repository(or else) select directory where you want to clone

git clone <repo url>

ls

....

now go to git command from vscode/cmd

git config --global --edit :will show user and password of current user if we want to change user make changes here

\*\*\*\*

vim <file name> :is a editor lear how to edit here

i :for insert

Esc :wq! or x! Enter :exit and save from editor

Exit without saving :q!

cat <filename> :check what changes has made

Note:->Learn command for vim editor

\*\*\*\*

**create new branch and make changes in it not in master**

git add <file name>

git commit -m "msg"

git push

now got to github can see make changes there

changes will show on fork not to master/owner

now got to 'contribute' option->open pull request->create pull request(before create, check what changes has made so far)->add comment-review again-create pull request

now master/owner will check what changes has made

owner will check->pull request->File Change->review change->approve..->review merge pull request -merge :now changes made in original repository ->

In commit it will show who committed .

So that multiple request can come and contribute in open source project

"Practise it, you will become a master in it."

-------------------------

git checkout rajkashy/multiply :we are in this repo

git status :if any changes

git add <filename>

git commit -m "msg" :committed in rajkashy/multiply

git push -u origin rajkashy/multiply :push to github

Note:-Repeat above for dev and master/main

------------------------

**If we want to work on others’ repository**

Fork(create repo and clone code) ->changes->pull request->approved by owner->submitted

-copy repository url

-got git command vscode

-create repository

-git clone <repo url>

-ls

-create new branch and make changes in it not in master

-git add <file name>

-git commit -m "msg"

-git push [(-u origin <branchname>)not require if has work once]