GIT - Version Control System (VCS)

- To track changes in files and folders
- To collaborate with team

  Types of VCS (2)

  - Centralised VCS
  - Distributed VCS

GIT is Distributed VCS - free and open source

GITHUB - website to upload repository online

- Provides backup
- Provides visual interface to repo
- Makes collaboration easier

GIT != GITHUB

For git check on mac command line

- Git --version or git -version // check the git is installed or not

Add github email name and username to git

- Git config --global user.email “email ac “
- Git config --global user.name “user name used in github”

Add files or folders to git - Tracking

- Cd /users/username/palce of folder / file name
- cd /Users/pramodrai/Documents/github/git/

To create a new git repo

- Git init

defaults write com. apple. finder AppleShowAllFiles YES. //to pop up hide folder

Git commit // used to save your changes to the local repository

Touch test.txt // used to create, change and modify timestamps of a file

Git add and files name // adds a change in the working directory to the staging area

Git commit -m “say something message” // m stands for message
Git add .   // adds a change in the working directory to the staging area
Git remote add origin   // add remote git into git terminal
Git push -u origin master // for the remote

Commands

(1)Git init
(2)Git status
(3)Git add
(4)Git commit -m “.................”
(5)Git remote add origin “location address”
(6)Git push -u origin master

Git log   // for the details of seeing everything
Git --help   // for the git help

Enable git commands autocomplete and colors on mac

Colors on mac terminal
To check colors

- Git config color.ui (if true, it is set or if No it is not set)

And then,
For setting color on mac terminal

- git config --global color.ui true

Branching and merging

Create branch
- Git branch “branch name”

Check out branch name
- Git checkout “branch name”

Merge new branch in master branch
- Git merge “branch name”

Delete branch
- Git branch -d “branch name” - delete from local
- Git push origin --delete “branch name”
-

  How to send Email from GitHub
  - Got to your repo
  - Go to settings
  - Go to Integration and Services
  - Go to add service
  - Add email services
  -

Tags ;
Git checkout master
Git tag <tab name> git tag v1.o
Display tag git show tag name
Push tags to remote == Git push origin tag name
Delete the tags = git tag -d tag name

Git merges vs git rebase

Git merging applies all unique commits from branch 1 into branch 2 in one commit with final result

1. How to install git
https://git-scm.com/downloads

2. how to create credentials on Github
https://github.com/

3. how to create repository on Github
click to create repository on https://github.com/

4. How to push existing project from local(own computer) to remote(github)
a. git init
b. git remote add origin <url>
c. git status
d. git add .
e. git status
f. git commit -m "initial project setup)
g. git push origin master

5. Config:

```
git config --global user.name "Coderwizard Tutorial"
```

```
git config --global user.email "test_coderwizard@gmail.com"
```

6. To View user.name and email

```
git config --list
```

7. For Help:

```
 git help
```

8. To Initialize a Repository

=> git init

9. To Add File To Staging Area

=> git add filename

=> git add filename1 filename2

=> git add . [period(.) is used to add all files]

10. To Commit Change

=> git commit -m 'short and meaningful comment '

=> git commit -am 'directly staging and Commit'

11. To clone Existing Repository

=> git clone <project_url>

12.To Check the Status Files

=> git status

=> git status -s [Short Status]

13. To view Difference

=> git diff [Find difference between committed and unstaged files]

=> git diff --staged [difference in staged files]

14. To Remove File From git

=> git rm <filename>

15. To Move File

=> git mv <FileName>
=> git mv file_from file_to [If we rename the file]

16. To Move staged file to not staged

```
=> git reset HEAD
 => git reset HEAD file1.txt
```

17. To Unmodify a Modified File

```
=> git checkout filename
```

18. To View Commit History

```
=> git log
=> git log --oneline [One line Easy and Fast]
=> git log --stat [More History]
 => git log --pretty=one line [Show SHA1 Value]
 => git log --pretty=format:"%h %s" --graph
=> git log --pretty=format:"%h - %an, %ar : %s"
=> git log --since=2.weeks => git log --after=2.weeks
 => git log --author "Coderwizard Tutorial"
 => git log --oneline --decorate --graph --all
```

19. To ignore files:

```
=> create .gitignore file
 => then add filename/foldername line by line
=> *.c => ignore al .c files
 => !main.c => not ignore main.c file
=> node_modules/
```

=> Directory node_modules and all files inside it.

20. To Unstage a Staged File with git restore

=> git restore

=> git restore --staged file2.txt

21. To Work With Remote Repository

=> git remote show origin [To View Remote Origin Details]

=> git remote -v

=> git remote add origin :

=> git push origin master

=> git pull

=> pull data with current head

=> git fetch

=> only download the data to local repository

=> git fetch origin mod

=> To retrive remote branch

=> git remote rename pb paul

=> git remote remove paul

=> git push origin --delete bad-branch-name

22. For Git Tag

=> git tag -l

=> git tag --list

```
   => git tag -a v1.4 -m "my version 1.4"

   => git show v1.4

   => git tag v1.4-lw [Lightweigt]

   => git tag -d v1.4-lw

   => git checkout v2.0.0
```

23. To Transfer Tag to Remote

```
   => git push origin .
```

24. Working With Branch

```
   => git branch testing [To Creating Testing Branch]

   => git checkout testing [Switch Branch]

   => git commit -am 'made a change'

   => git checkout master [To Move to Master Branch]

   => git checkout -b .[Creating branch and switching to it]

   => git branch --merged

   => git branch --no-merged

    => git branch --move bad-branch-name corrected-branch-name
```

25. Redo Commit

```
   => git reset

   => git reset SHA1

   => git reset -hard SHA1
```

26. For Git Stash

```
=> git stash save

 => git stash list

=> git stash pop

=> git stash apply stash{0}

 => git stash clear

=> git stash grop stash
```

27. To Merge

```
git merge <branch name>

 git merge --no-ff branch name
```

Rebase

```
git rebase <branch name>
```