1
1
Yesterday Topics
---------- 1.
What is the need of VCS?
2. How many types of VCS?
a. LVCS
b. CVCS eg: SVN
c. DVCS eg: GitHub,bitbucket,gitlab,.....
3. Architecture of git
a. working area => devleoper workspace(source code)
b. Stage area => Before adding the code to the localrepository, we keep the code in stage area
c. Local repository => Before pushing the code to remote repository, we keep the code in "Local repository".
4. What is git and github?git -> It is a client tool where the user will enter url,username and password of github repository. github-> It is a server where we keep repositories/projects which would be used for colloboration.

git commands(case sensitive)
==========
1. git version
2. git help
3. git config
4. git init
5. git clone
6. git add
7. git status
8. git rm
9. git restore
10. git commit
11. git log
12. git show
13.git push
14. git pull
15. git branch
16. git checkout
17. git stash

Note: git merge, git rebase

1. git versionThis command is used to check the version of git

syntax: git version or git --version

nitin@DESKTOP-1N5U4UJ MINGW64 ~
$ git version
git version 2.37.0.windows.1

nitin@DESKTOP-1N5U4UJ MINGW64 ~
$ git --version git version
2.37.0.windows.1

2. git help If we want to see the list of commands then we can use
git help syntax : git help

Note: This command is useful to get the documentation of any command
eg: git help <command-name>

3. git configIt is used when the git software is used for the first time.
The command will set the devleoper identity like name,emailid,....
This configuration information will be used by git software for every push operation encounterd.

> git config --list //this command is used to provide the list of configuration

```
$ git config --list
diff.astextplain.textconv=astextplain filter.lfs.clean=git-
lfs clean -- %f filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true core.fscache=true core.symlinks=false
pull.rebase=false credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
```

//to set the username and email

```
> git config --global user.name  "Nitin"
> git config --global user.email "javabynitin2022@gmail.com"
```

global => it indicates the user can work with git commands from different drives of computer.

Note: git config --list --show-origin //display the location of git configuration holded by git software

Important operations associated with git
==========

git init

=> normally a folder will be created in the developers works place and inside the folder the source code would be place
=> normally this is the first command which we execute to set up the git for operations like
clone,push,pull,....
=> This command internally creates one folder called .git
=> .git is used by git software to identify the folder which should participate in pushing to "local" and
"remote" repositories.

syntax : git init

Note: To change the directory use the following command cd .....
To check in which directory we are currently present we use pwd command

git status

> This command is used to check the status of the working directory
> git status

D:\gitsession\Workspace-1(master)
                          |=> Demo.java

On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Demo.java
nothing added to commit but untracked files present (use "git add" to track)

 git status normally will give outputs in the following ways
              a. untracked files(red color) => it means the files are present still in working area and these files can't be commited to "local repository" nor to "remote repository"
                 b. tracked files(green color) => it means the files are moved from working area to stage area so these files can be commited to "local repository" and to
                                                "remote repository".
            c. modified files(red color) => it means the files are present still in
                                working area and these   files can be staged or it can also be restored back to the   normal phase.

To send the code from workspace to stagearea we use the following command
            syntax : git add <file-name>
If we want to push all the files from workspace to stage area,we use the following command syntax: git add . git add --a

It is also possible to unstage the files from stagged area to workspace,using the following command
           syntax: git rm --cached <file-name>

To restore the old file we use the following command
syntax: git restore <file-name>

The files which are ready for commit should be in stage area, to perform commit operation we use the following command   syntax: git commit -m <some-messages>

      eg#1. git commit -m "first commit"   //This file commit all the files present in stage area
       eg#2. git commit -m "second commit" filename //This will commit only that file into local repository

Steps followed to create a remote repository and push it to remote repository
----
----
1. open github.com by providing the credentials
2. create a new repository and enter some name(repository name) and click on
    create repository
3. To peform push operation we need to use the following command git branch -M
   main
      git remote add origin https://github.com/NitinTechnology/Workspace.git
      git push -u origin main

Difference between pull and clone
git pull -> it is used to fetch the lastest changes made in remote repository to working directory. syntax: git pull

git clone-> it is used to clone the repository to the working directory of the devleoper. syntax: git clone <url>