1. What is Git?
   1. Decentralized/Distributed
   2. Can be centralized
   3. Created for Linux Kernel Project—contrubuted more than around 12000 devlopers and contains over 50 million lines of code
2. De-facto standard- wide adoption and integration
3. The Repository
   1. Collection of Files Managed by Git
   2. History (all of it)
   3. Working Directory/ Workspace
   4. Quasi-hidden .git Folder
   5. Tip: Focus Purpose of Repository
4. Commits and Files
   1. Snapshots are commits
   2. One or More Files Changes
   3. Dummy File to “save” a Folder
   4. Commits on Timeline (Branch)
5. Git for Windows
   1. Minimal Configuration required
      1. In Git Bash: git config --global user.name “manmath markunde”
      2. In Git Bash: git config --global user.email “[manmath738@gmail.com](mailto:manmath738@gmail.com)”
6. Make Notepad++ as default editor in gitbash
   1. In Git Bash: notepad++ ~/.bash\_profile
      1. inside .bash-profile file add new line as
         1. In Git Bash: alias npp=’notepad++ -multiInst -nosession’ then restard the gitbash
      2. Now set Default text editor in Git us following command
         1. In Git Bash: git config --global core.editor “notepad++ -mulitInst -nosession”
      3. To test Git integration, by using command if the notepad++ opens git default configuration file then configuration successfull.
         1. In Git Bash: git config --global -e
7. P4Merge for Windows ([http://perforce.com](http://perforce.com/) & option- HELIX P4MERGE: VISUAL MERGE TOOL
   1. Visual Compare and merge Tool
   2. Command line Merging is annoying
   3. this is an Optional- But Highly recommended
      * 1. In Git Bash: git config --global diff.tool p4merge
        2. In Git Bash: git config --global difftool.p4merge.path “[C:/Program](../../C:/Program) Files/Perforce/p4merge.exe”
        3. **Troublishooting: Try replacing** [**C:/**](../../C:/) **with /*c*/ instead**
        4. In Git Bash: git config --global difftool.prompt false
        5. P4Merge as graphical Merge Tool
           1. In Git Bash: git config --global merge.tool p4merge
           2. In Git Bash: git config --global mergetool.p4merge.path “[C:/Program](../../C:/Program) Files/Perforce/p4merge.exe”
        6. In Git Bash: git config --global mergetool.prompt false
        7. git version
8. Auto create demo folder and within this folder git generates .git folder

In Git Bash: git init demo

* + - 1. alternate, using bash go to the demo directory and use command to generate .git folder

In Git Bash: git init . ["." represents current directory]

1. README.md
   * + 1. The ".md" extension is for markdown, which is a popular file format.
2. Git Staging
   * + 1. staging is used to prepare for next commit

In Git Bash: git add "fileName"

* + - 1. this will add untracked working directory into git staging area, to verify that the file is in staging area by using command

In Git Bash: git status

* + - 1. Now,the file will show status as - unstage and changes to be commited, for that use command

In Git Bash: git commit -m "Commiting Message"

* + - 1. after that if we check again git status command then it will show that, repository on branch master and nothing to commit, working directory clean
      2. by mistake user just used command "git commit" then a default text editor will open and user has to put commit message on that file

1. View all commits log and show

In Git Bash: git log

* + - 1. command will show all the commit messages of current repository/branch In Git Bash: git show
      2. command will show the last commit and a diff containing all the changes.

1. command will show only the tracking files which are being track by git.

In Git Bash: git ls-files

1. The express command
   1. which will first add modified file into staging and then commits message, the express command can't work for untracked files

In Git Bash: git commit -am