**GIT :- Is the best version control system.**

Git allows groups of people to work on the same documents (often code) at the same time, and without stepping on each other's toes. It's a distributed version control system.

Download the latest git from the url :- <https://git-scm.com/downloads>

Chose the default configurations, and click next.

Recommended :- Use git from Git Bash Only

Commands:-

git --help

git init :- which initialize the repository the local folder

which creates a “.git” folder which contains the change set related info, any time you can delete this “.git” to remove the project from the repository.

status command:- it helps you to shows the current changes to the repository like files modified, files added, files deleted.

see what the current state of our project is.

>git status

if it sees any new file, git says the file is "untracked"?

if it sees “changes to be committed?” The files listed here are in the Staging Area, and they are not in our repository yet. We could add or remove files from the stage before we store them in the repository.

git log:- gives you the commit history

git add ‘\*.\*’

git add ‘\*.html’

dont forgot to add the quotes.

To ignore certains file to be added to repository

in the same folder, add .gitignore file, and keep the file names for ex:- \*.log

commit command:- It remembers the saved point,and any time you can revert to previous point, and also Its makes ready your file to be pushed to repository.

git commit -m "Add cute octocat story"

Push command:- to push our local changes to the remote repository, -u is to remember the parameters used so that next time you can use ‘git push’ directly.

git push -u origin master

pull command:- to pull other changes to local.

git pull origin master

rm 'blue\_octocat.txt'

git reset HEAD "\*.\*" //to remove from add state

git diff xyz.txt //gives the difference with last commit

**git push**

**userName: userName**

**password: password**

**changes will be published to remote repo.**

**Getting your project on Git-Hub**

References:-

<https://guides.github.com/introduction/getting-your-project-on-github/#pullit>

<https://github.com/features>

<https://try.github.io>

https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository

**Step by step guid to setup github:-**

1.Signup<https://github.com/>

2.Download Github for windows and install it

**Github is the place where you can keep your project online** for collabaration,

Other people can download your code, and make changes and push to the public repo.

GIT : - Git best version control tool in the industry currently.

->Share project “ExampleProject”, for example 100 files there.

-> 10 persons working on the same project.

->two persons working on the same file “myfile.txt”.

→ You wanted to convert your project into the GIT, for making use of the GIT Version control tool.

Download the latest git from the url :- <https://git-scm.com/downloads>

Chose the default configurations, and click next.

Recommended :- Use git from Git Bash Only.

1.You want to create a share project, you wanted to GIT version control for your project.

### [Cloning an Existing Repository](https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository#Cloning-an-Existing-Repository)

If you want to get a copy of an existing Git repository – for example, a project you’d like to contribute to – the command you need is git clone. If you’re familiar with other VCS systems such as Subversion, you’ll notice that the command is “clone” and not “checkout”. This is an important distinction – instead of getting just a working copy, Git receives a full copy of nearly all data that the server has. Every version of every file for the history of the project is pulled down by default when you run git clone. In fact, if your server disk gets corrupted, you can often use nearly any of the clones on any client to set the server back to the state it was in when it was cloned (you may lose some server-side hooks and such, but all the versioned data would be there – see [“Getting Git on a Server”](https://git-scm.com/book/en/v2/ch04/_git_on_the_server) for more details).

You clone a repository with git clone [url]. For example, if you want to clone the Git linkable library called libgit2, you can do so like this:

$ git clone https://github.com/libgit2/libgit2

That creates a directory named “libgit2”, initializes a .git directory inside it, pulls down all the data for that repository, and checks out a working copy of the latest version. If you go into the new libgit2 directory, you’ll see the project files in there, ready to be worked on or used. If you want to clone the repository into a directory named something other than “libgit2”, you can specify that as the next command-line option:

$ git clone https://github.com/libgit2/libgit2 mylibgit

That command does the same thing as the previous one, but the target directory is called mylibgit.

Git has a number of different transfer protocols you can use. The previous example uses the https:// protocol, but you may also see git:// or user@server:path/to/repo.git, which uses the SSH transfer protocol. [“Getting Git on a Server”](https://git-scm.com/book/en/v2/ch04/_git_on_the_server) will introduce all of the available options the server can set up to access your Git repository and the pros and cons of each.