**Name:** [Naveenraj Karthikeyan](mailto:naveenraj.karthikeyan@relevantz.com) **Emp Id:** 11958

**Self Preparation:**

**Git and Github Tutorial:**

**(08-08-2023) - self preparation.**

**Git:**

**Git** is an **open-source distributed version control system**. It is designed to handle minor to major projects with high speed and efficiency. It is developed to coordinate the work among the developers. The version control allows us to track and work together with our team members at the same workspace.

**Features of Git:**

* Open source
* Scalability
* Speed
* Security

**Benefits of Git:**

* Saves time
* Offline working
* Undo mistakes
* Track changes

**Github:**

GitHub is an online hosting service for Git repositories. Imagine working on a project at home and while you are away, maybe at a friend's place, you suddenly remember the solution to a code error that has kept you restless for days.

You cannot make these changes because your PC is not with you. But if you have your project hosted on GitHub, you can access and download that project with a command on whatever computer you have access to. Then you can make your changes and push the latest version back to GitHub.

**Features of Github:**

GitHub is a place where programmers and designers work together. They collaborate, contribute, and fix bugs together. It hosts plenty of open source projects and codes of various programming languages.

Some of its significant features are as follows.

* Collaboration
* Integrated issue and bug tracking
* Graphical representation of branches
* Git repositories hosting
* Project management
* Team management
* Code hosting
* Track and assign tasks.

**Git version control system:**

A version control system is a software that tracks changes to a file or set of files over time so that you can recall specific versions later. It also allows you to work together with other programmers.

**For example,** you could be working on a website's landing page and discover that you do not like the navigation bar. But at the same time, you might not want to start altering its components because it might get worse.

With Git, you can create an identical copy of that file and play around with the navigation bar. Then, when you are satisfied with your changes, you can merge the copy to the original file.

You are not limited to using Git just for source code files – you can also use it to keep track of text files or even images. This means that Git is not just for developers – anyone can find it helpful.

**Intro to Git - Udemy: ( 09-08-2023)**

**What is Git?**

* Git is a version control system that stores the reference points to snapshots of a code.
* This creates a linear timeline of all your changes which allows you to go back in time to earlier snapshots in the event that you messed up your current code or need to look at something that you did previously.

**Git commands:**

**ls -** Shows the list of files in the directory.

**mkdir -** Make a directory.

**Cd -** Change directory.

**$ git init -** used to initialize the git repository. By default git branch is **master**.

**$ touch <file> -** used to create, modify and delete files.

**$ git status -** used to track the status of what is happening in the backend of each commit.

**$ git add -** used to add the file in the repository.

**$ git commit -m “Commit message” -** will commit the file inside the staging area.

**$ git log -** will show the nice version histories we have made in the git repository.

**Adding all the files of similar type to staging area:**

**$ git add \*.html -** It will add all the same file type files to the repository.

**Adding all files including hidden files:**

**$ git add -A (or) - - all -** Adds all files and folders from the directory you're in.

This is a good command for adding everything in your project all at one

time.

**Ignoring files and folders:**

**$ touch .gitignore -** It will ignore the files that are untracked. For that we want to specify the filename inside the .gitignore file.

**Branches:**

Branches are used to create new features of a particular branch. By creating a new branch we can implement the new feature without affecting the master branch.

Once we do a new feature we are able to merge them in the old branch.

**Listing all branches:**

**$ git branch -** List all the branches inside the repository.

**Adding a branch:**

**$ git checkout -b <branch name> -** It will change the master branch to a new branch.

**Checking out a branch:**

**$ git checkout <Branch name> -** One branch to another branch.

**Merge a branch:**

**$ git merge <Branch name> -** It will merge the new branch to the old branch.

**Remove a branch:**

**$ git branch -d <Branch Name> -** Delete or remove the branch.

**Moving one directory files to another directory:**

**$ mv <file name> <folder name> -** It will move the file from one folder to another folder.

**Renaming the file:**

**$ mv <file name> <file name>**

**Intro to Github:**

1. **Create a Github Account.**
2. **Make a new Repository.**
3. **Copy the path and add to the git command.**

**Eg:** $ git remote add origin <https://github.com/Naveenrajk/Intro-git.git>

**$ git remote -v -**It says the origin of the push and fetch path.

**$ git push -u origin master -** It will add all the files to the remote repository.

**“These commands all are practiced in the system as well as added to the local repository to the remote using github”.**

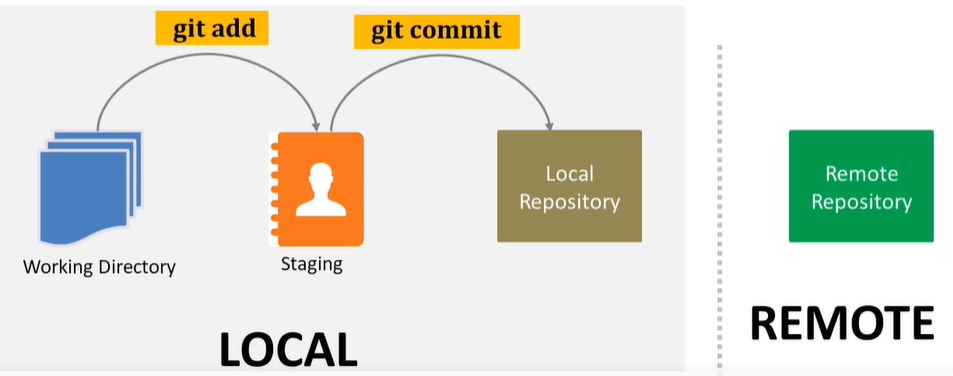
**Git tutorials for Beginners - Udemy: ( 09-08-2023)**

**Git Commands:**

**$ git init -** used to initialize the git repository. By default git branch is **master**.

**$ git clone -** used to clone a repository from remote to local.

Eg: $ git clone <url>



**$ git add -** used to add the file in the repository.

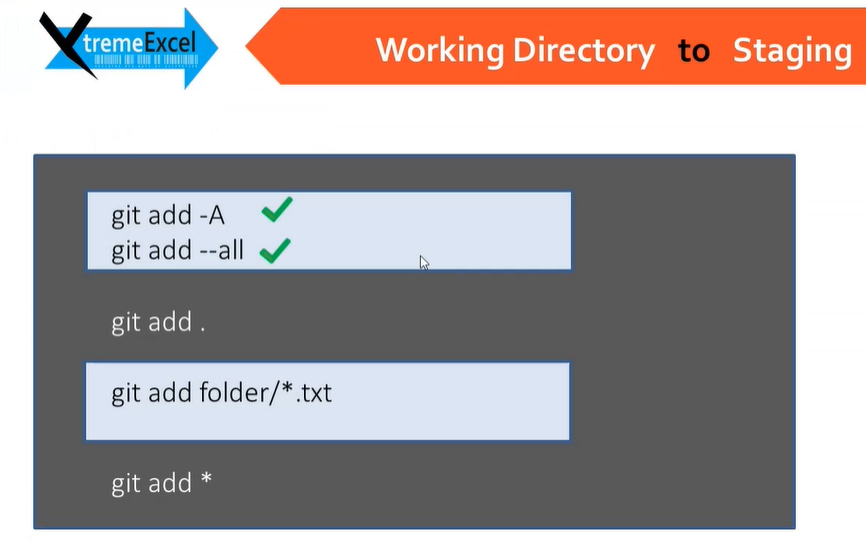
**$ git commit -m “Commit message” -** will commit the file inside the staging area.

**ADDING WORKING DIRECTORY TO STAGE:**

**$ git add -A (or) - - all -** Adds all files and folders from the directory you're in.

This is a good command for adding everything in your project all at one

time.





**$ git diff <filename> -** used to check the differences made inside the particular file.(working directory)

**$ git diff - - cached <filename> -** to be used while the file is in the staging area.

