**Nishankumar M**

# Git & GitHub

## Git:

- Git is a software used to track files and code changes.
- Git provides an isolated area to develop the project.

## Repository:

- Repository is a isolated area to develop the project.
- It contains folders, files, images, videos, spreadsheets, and data sets whatever you need for the project.
- It is usually used to organize a single project.
- It includes a README.md file, a file with information about your project.
- We can create a branch too.

## Branches:

- Branch allows you to develop features, for bugs or safely experiment with new ideas in a contained area of your repository.

## Forking a repository:

- Forking is making a copy of a repository.
- Forking a repository allows you to freely experiment with changes without affecting the original project.

## Git Configuration Level:

- Three levels of configuration in Git.
    - System level
    - Global level
    - Local level

Note: Local overrides Global and Global overrides System.

**1 step of Git configuration:**

- **User.name and user.email**

1. **Need to define a level user.name**

```
git config –global user.name “Name”
```

2. **Need to define user.email**

```
git config –global user.email “Email”
```

3. **Configure default editor**

```
git config –global core.editor “editor installation path upto .exe file”
```

Eg: c:\Program Files\Sublime_text\Sublime_text.exe

4. **View information related to no. Of configurations execute**

```
git config –list
```

5. **To see the username and user email of the git**

```
git config –global user.name
git config –global user.email
```

## Steps to create a Git Repository:

--> Create an empty directory

```
mkdir <directory-name>
```

--> Convert the directory into repository

```
git init
```

--> Check status

```
git status
```

Note: git init - creates a hidden directory that stores all the metadata required for it to function.

--> Check the contents of the directory

```
Ls –la
```

## Git Work Flow

- Git work flow has three stages.
    - Stage 1 - Working directory
    - Stage 2 - Staging area
    - Stage 3 – Local repository

**Working directory -------------> Staging area ---------------> Local repository**

[untracked files]        git add .                        git commit -m “msg”

Note: -m flag indicate to give a message for commit

**Managing and viewing changes:**

- Whenever the commit is happened it provide us 4 details of commit
    - Commit id
    - Author name
    - Commit date
    - Commit message
- To view the commit details use the following command:

git log – it displays all the committed details line by line

- To view commit details in one line use following command:

git log --oneline – it displays the commit details in single line

## Track Changes Between the Stages

Tracking is the ability to identify the changes between the different versions of the same file, spread across various repositories.

- Track changes between the working directory and staging area:

```
git diff <filename>
```

- Track changes between the staging area and local repository:

```
git diff –staged
```

Note:

- git diff –staged similar to git diff HEAD
- If number of commits exceed, use hash code

```
git diff <commit hash code>
```