Git

* Git is a version control system.
* Git helps you keep track of code changes.
* Git is used to collaborate on code.

What is Git?

Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

It is used for:

* Tracking code changes
* Tracking who made changes
* Coding collaboration

What does Git do?

* Manage projects with **Repositories**
* **Clone** a project to work on a local copy
* Control and track changes with **Staging** and **Committing**
* **Branch** and **Merge** to allow for work on different parts and versions of a project
* **Pull** the latest version of the project to a local copy
* **Push** local updates to the main project

Working with Git

* Initialize Git on a folder, making it a **Repository**
* Git now creates a hidden folder to keep track of changes in that folder
* When a file is changed, added or deleted, it is considered **modified**
* You select the modified files you want to **Stage**
* The **Staged** files are **Committed**, which prompts Git to store a **permanent** snapshot of the files
* Git allows you to see the full history of every commit.
* You can revert back to any previous commit.
* Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

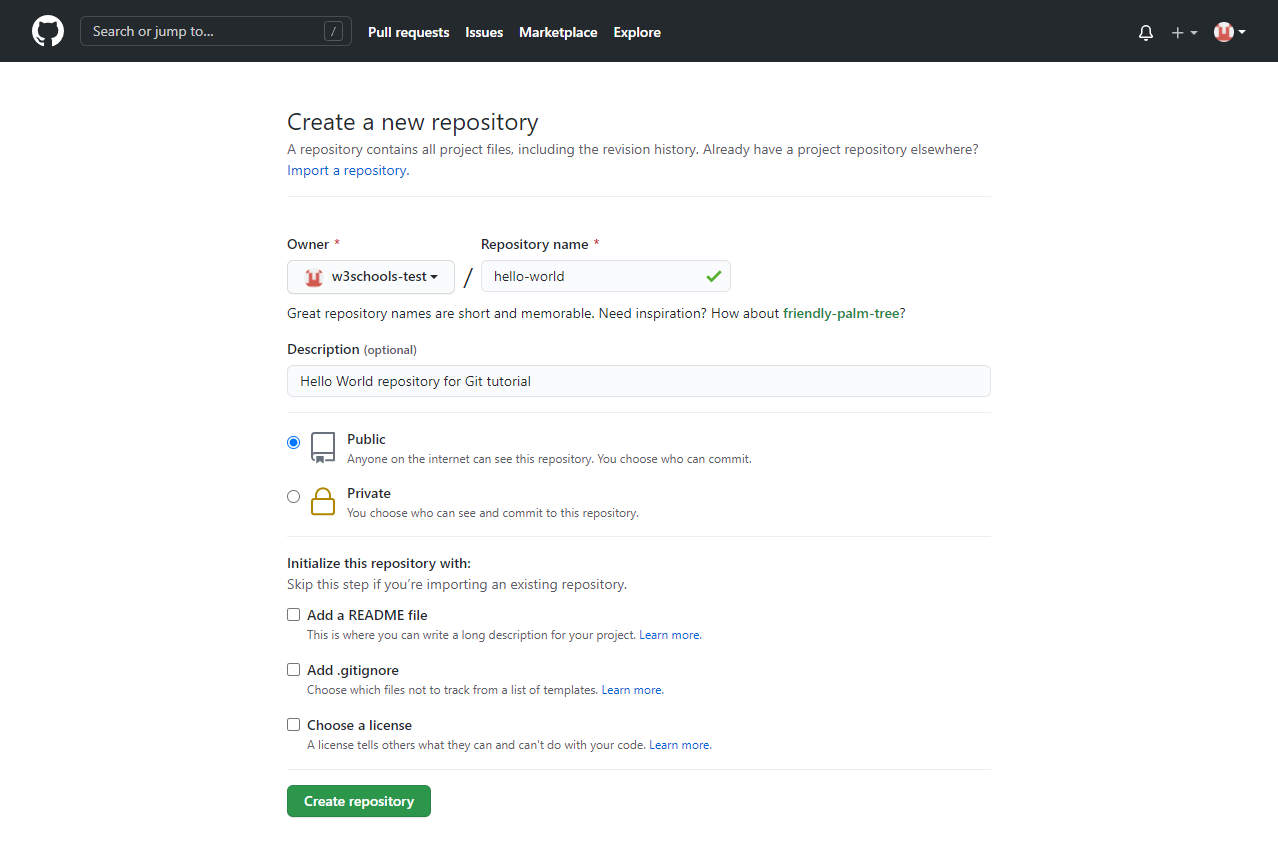
Why Git?

* Over 70% of developers use Git!
* Developers can work together from anywhere in the world.
* Developers can see the full history of the project.
* Developers can revert to earlier versions of a project.

What is GitHub?

* Git is not the same as GitHub.
* GitHub makes tools that use Git.
* GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018.
* In this tutorial, we will focus on using Git with GitHub.

## Create a Repository on GitHub



Git

* 1. Install git

<https://git-scm.com/downloads>

All Command Git :

1. git config --list
2. git config --global user.name “Raju Ranjan Kushwaha”
3. git config --global user.email [ranjanrajukumar4@gmail.com](mailto:ranjanrajukumar4@gmail.com)
4. git config --list

{ First install git then setup username and email of the verify git on your system}

* 1. git status

{This command is this current folder status in git repository}

Push Code on git:-

1. git status
2. git init
3. git status
4. git add.
5. git commit –m “message”
6. git push –u origin master [URL ON ROPOSITORY]

### -------------------------------------------------------------------

#### -----------------------------------------------------------------------------------

### Local Repository

### Remote Repository

#### ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

* 1. git log
  2. git log –oneline

## Create Branching on git:-

## HAS VALUE[ NO Command]

## Command:-

## git log –oneline

## git branch branchname Has value

## git branch[all see branch]

## git checkout Branchname

## git add.

## git commit –m “message”

## git push –u origin Branchname

## -----------------------------------------------------

## Merge:-

## get checkout master

## git log

## git branch

## git merge Branchname

## git push –u origin master

## [ i press -> Insert ,ESC :Wq[write quite]

## ------------------------------------------------------------------

## Delete Branch:-

## git branch

## git branch –d branchname

## git push –d origin Branchname/Master

## Config:-

## >>>>>master

## <<<<<<branch

## Stash :-

## git stash –u

## git stash list

## git stash apply

## ---------------------------------------------------------

## [Uncomplete code then change Branch to mastar]

## ==========================

1.git stash save “message”

# 2.git stash save –u [or –all]

3.git stash show

4. git stash show –p

5.git stash pop[ delete stash]

IGNORE FILE:-

* 1. touch gitignore(create file)
  2. mkdir dir1

/file.text

/dir1

Staging Area :-

1. git commit –a –m “skkiped staged area”

Git Diff:-

1. git diff
2. git diff –staged
3. git diff –cached

[showing changes b/w WD & SA ,WD & LR ,SA & LA]

Rename & Remove

1. git mv file4.txt file3.txt[rename]
2. git rm file3.txt[Delete]
3. git rm –cached file.txt

Restore:-

1. git restore file.txt

Reset & Revert:-

1. –hard
2. –soft
3. –mixed

1.git log –onelist

2.git reset –hard SHA value

3.git push –u origin master

4. git push –f –u origin master

5.git clean -n