



Dr. Vishwanath Karad

**MIT WORLD PEACE  
UNIVERSITY** | PUNE

TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

School of Computer Engineering and Technology  
Academic Year: 2023-2024 Sem V  
**Full Stack Development (FSD)**

**Lab Assignment : 01**

**Title:** Version control with Git.

**Prepared By**  
Saurabh Jitendra Jadhav

Batch A1  
September 20,2023

**Aim:-** Version control with Git.

**Objectives:**

1. To introduce the concepts and software behind version control, using the example of Git.
2. To understand the use of 'version control' in the context of a coding project.
3. To learn Git version control with Clone, commit to, and push, pull from a git repository.

**Theory:**

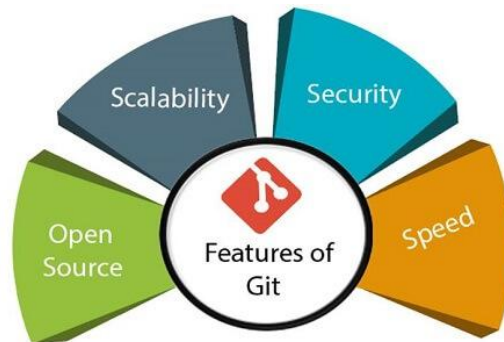
1. What is Git? What is Version Control?

->

- ❖ 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 co-ordinate the work among the developers**. The version control allows us to track and work together with our team members at the same workspace.
- ❖ Git was created by **Linus Torvalds in 2005** to develop Linux Kernel. It is also used as an important distributed version-control tool for the DevOps.
- ❖ 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.
- ❖ The version control system is a collection of software tools that help a team **to manage changes in a source code**. It uses a special kind of database to keep track of every modification to the code.
- ❖ **Features:-**

**1. Open Source:-**

Git is an open-source tool. It is released under the GPL (General Public License) license.



## 2. Scalable:-

Git is scalable, which means when the number of users increases, the Git can easily handle such situations.

## 3. Distributed:-

One of Git's great features is that it is distributed. Distributed means that instead of switching the project to another machine, we can create a "clone" of the entire repository. Also, instead of just having one central repository that you send changes to, every user has their own repository that contains the entire commit history of the project.

## 4. Security

Git is secure. It uses the SHA1 (Secure Hash Function) to name and identify objects within its repository. Files and commits are checked and retrieved by its checksum at the time of checkout.

## 2. How to use Git for version controlling?

->Using Git for Version Control

### 1. Install Git:

- Download and install Git from the official website (<https://git-scm.com/>).

### 2. Configure Git:

- Set your identity using the following commands, replacing "Your Name" and "your.email@example.com" with your actual name and email address:

```
git config --global user.name "Your Name"  
git config --global user.email "your.email@example.com"
```

### 3. Create a Git Repository:

- Initialize a Git repository in your project's directory:

```
git init
```

### 4. Add and Commit Changes:

- After making changes to your project files, stage and commit them with a message:

1. 

```
git add .
```

 # Stage all changes

2. 

```
git commit -m "Initial commit"
```

# Commit changes with a descriptive message

### 5. Check Status and History:

- To check the repository status and view the commit history, use:

```
git status
```

```
git log
```

### 6. Create and Switch Branches:

- Create a new branch and switch to it:

```
git branch new-feature
```

 # Create a new branch

```
git checkout new-feature # Switch to the new branch
```

## 7. Merge Branches:

- Merge a completed branch back into the main branch (e.g. master):

```
git checkout master # Switch to the main branch  
git merge new-feature # Merge the new feature branch into master
```

## 8. Resolve Conflicts:

- If conflicts occur during a merge, manually resolve them in the affected files and commit the changes.

## 9. Push to Remote Repository:

- Push changes to a remote repository (e.g., GitHub or GitLab) when collaborating with others:

```
git push origin master # Push changes from the local master branch  
to the remote master branch
```

## 10. Pull from Remote Repository:

- Get the latest changes from the remote repository:

```
git pull origin master # Pull changes from the remote master  
branch to the local master branch
```

These are the fundamental Git commands and workflows for version controls.

## Implementation :-

Github Repository Link:- <https://github.com/Saurabh3207/calculator>

### A) Initializing local Repository:

```
➔ Calculator_Project pwd
/Users/saurabhbalsahahebajadhav/Documents/Calculator_Project
➔ Calculator_Project git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/saurabhbalsahahebajadhav/Documents/Calculator_Project/.git/
➔ Calculator_Project git:(master) git status
On branch master

No commits yet
```

### B) Adding files to Repository

```
➔ Calculator_Project git:(master) x git add .
➔ Calculator_Project git:(master) x git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   index.html
    new file:   script.js
    new file:   style.css
```

### C) Initial Commit.

```
➔ Calculator_Project git:(master) x git commit -m "Initial commit"
[master (root-commit) 1612e39] Initial commit
3 files changed, 59 insertions(+)
create mode 100644 index.html
create mode 100644 script.js
create mode 100644 style.css
```

```
commit 1612e390c413968bb13037797354edcd5e7ab5e0 (HEAD -> master)
Author: Saurabh3207 <1032210970@mitwpu.edu.in>
Date: Mon Sep 18 11:48:56 2023 +0530
```

Initial commit

(END)

## D) Pull Request.

```
➤ Calculator_Project git:(main) ✗ git pull origin main

remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 12 (delta 6), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), 3.04 KiB | 222.00 KiB/s, done.
From https://github.com/Saurabh3207/calculator
* branch      main      -> FETCH_HEAD
   605eb8c..b3d772a main  -> origin/main
Updating 605eb8c..b3d772a
Fast-forward
 README.md | 15 ++++++++
 index.html | 6 +---
 2 files changed, 17 insertions(+), 4 deletions(-)
```

## E) Push Request.

```
➤ Calculator_Project git:(main) ✗ git push origin main

Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.03 MiB | 15.24 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Saurabh3207/calculator.git
   a9009c5..605eb8c main -> main
```

## D) Adding Images Directory to local Repository.

```
➤ Calculator_Project git:(main) ✗ git add Output_Images/
➤ Calculator_Project git:(main) ✗ git status

On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   Output_Images/Calculator_Light.png
    new file:   Output_Images/Calculator_dark.png

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .DS_Store
```

## E) Linked Local Repository to Remote Repository.

```
➤ Calculator_Project git:(main) ✗ git remote add origin https://github.com/Saurabh3207/calculator.git
➤ Calculator_Project git:(main) ✗ git push -u origin main
```

## Github Repository:-

The screenshot shows the GitHub interface for a repository named 'calculator' owned by 'Saurabh3207'. The repository is public and has 0 stars, 0 forks, and 1 watcher. The main branch is 'main'. The repository contains several files and folders: 'Output\_images', 'README.md', 'index.html', 'script.js', and 'style.css'. The most recent commit is 'Update index.html' by 'Saurabh3207' 2 days ago. The repository description is 'Simple calculator using html ,css and js'.

**Repository Details:**

- Repository: calculator (Public)
- Owner: Saurabh3207
- Stars: 0
- Forks: 0
- Watching: 1
- Activity: 0
- Readme: 1

**Files and Commits:**

File/Folder	Description	Commit	Time
Output_images	Add output_images folder with screenshot	b3d772a	2 days ago
README.md	Update README.md		2 days ago
index.html	Update index.html		2 days ago
script.js	Deleted some unwwanted files from project directory		2 days ago
style.css	Deleted some unwwanted files from project directory		2 days ago

**Conclusion:-** Thus We learn and implement Assignment based on Git version. We successfully created Github Repository.