

# Introduction to Git

## ■ Definition

- Git is a distributed version control system (VCS) designed to handle projects of all sizes with speed and efficiency.

## ■ Purpose

- Tracks changes to source code over time.
- Facilitates collaboration among developers.
- Manages project versions and history efficiently.
- Real-time Example: In a team project, multiple developers push their changes to a shared GitHub repository while maintaining history.

## ■ Key Features of Git

- Distributed System: Every developer has a full copy of the repository.
- Performance: Optimized for speed and efficiency.
- Branching and Merging: Lightweight branching and merging to support parallel development.
- Data Integrity: Ensures data is not corrupted during transit or storage.
- Support for Non-linear Development: Ability to work on multiple tasks simultaneously and merge changes as needed.
- Real-time Example: A developer creates a feature branch to test a new login system while another developer works on UI updates, then both merge into main without conflicts.

## ■ Comparison: Git vs Traditional VCS

Aspect	Traditional VCS	Git
Repository Model	Centralized (one main server)	Distributed (every user has full copy)
Offline Work	Limited or not possible	Full functionality offline
Branching	Heavyweight and costly	Lightweight and fast
Performance	Dependent on server availability	Optimized for speed
Data Integrity	Single point of failure	Cryptographic integrity checks

## ■ Conclusion

- Git is the most widely used distributed version control system in modern software development.

- Its distributed nature, powerful branching, and data integrity make it ideal for collaborative development.
- By enabling both individual productivity and team collaboration, Git has become a standard tool across industries.