**NB:** [**resource to use**](https://www.w3schools.com/git/default.asp)

**1. Introduction to Version Control**

* Definition and Importance
* Types of Version Control Systems (Centralized vs. Distributed)
* Overview of Git and its Features

**2. Getting Started with Git**

* Installing Git
  + Installation on Windows
  + Installation on macOS
  + Installation on Linux
* Basic Configuration
  + Setting up username and email
  + Configuring default text editor
  + Configuring line endings

**3. Git Basics**

* Understanding the Git Workflow
  + Working Directory
  + Staging Area
  + Repository
* Creating a Repository
  + Initializing a new repository
  + Cloning an existing repository

**4. Making Changes**

* Basic Commands
  + git status
  + git add
  + git commit
  + git log
* Tracking Changes
  + Viewing changes with git diff
  + Viewing commit history with git log
* Undoing Changes
  + Unstaging files with git reset
  + Amending commits with git commit --amend
  + Reverting changes with git revert

**5. Branching and Merging**

* Understanding Branches
  + Creating branches with git branch
  + Switching branches with git checkout
* Working with Branches
  + Merging branches with git merge
  + Resolving merge conflicts
  + Deleting branches
* Advanced Branching
  + Rebasing with git rebase
  + Cherry-picking with git cherry-pick

**6. Remote Repositories**

* Working with Remotes
  + Adding a remote repository with git remote add
  + Fetching changes with git fetch
  + Pulling changes with git pull
  + Pushing changes with git push
* Collaboration Workflow
  + Forking a repository
  + Creating pull requests
  + Reviewing and merging pull requests

**7. Advanced Git**

* Stashing Changes
  + Stashing with git stash
  + Applying stashes
  + Dropping stashes
* Tagging
  + Creating tags with git tag
  + Annotated vs Lightweight tags
  + Listing and deleting tags
* Git Hooks
  + Overview of Git hooks
  + Setting up and using hooks

**8. Best Practices**

* Writing Good Commit Messages
* Branching Strategies (e.g., Git Flow, GitHub Flow)
* Keeping a Clean Commit History
* Using .gitignore effectively