# Git and GitHub Hands-On Exercises

## What is Git?

Exercise: Define Git and list three benefits of using it.  
  
Solution:  
Git is a distributed version control system that allows multiple developers to collaborate on a project.  
  
Benefits:  
1. Tracks changes in the code.  
2. Enables collaboration with others.  
3. Allows branching and merging for parallel development.

## Install Git on Windows

Exercise: Install Git on your Windows system.  
  
Solution:  
1. Go to [Git's official website](https://git-scm.com).  
2. Download the installer for Windows.  
3. Run the installer and follow the setup instructions, selecting appropriate options.  
4. Verify the installation using the command:  
 git --version

## What is GitHub?

Exercise: Create a GitHub account and explore its interface.  
  
Solution:  
1. Visit [GitHub](https://github.com) and sign up for a free account.  
2. Explore key features like repositories, pull requests, issues, and GitHub Actions.

## Git vs. GitHub

Exercise: Differentiate Git and GitHub with examples.  
  
Solution:  
Git: A version control system for tracking changes. Example:  
 git init  
  
GitHub: A hosting service for Git repositories. Example: Hosting a repository and sharing code.

## What is GitLab?

Exercise: Sign up on GitLab and create your first project.  
  
Solution:  
1. Go to [GitLab](https://gitlab.com).  
2. Sign up and verify your email.  
3. Create a new project using the web interface.  
4. Clone it using:  
 git clone <your-repo-URL>

## Git Initialization

Exercise: Initialize a Git repository locally.  
  
Solution:  
 mkdir my-project  
 cd my-project  
 git init

## Git Status

Exercise: Check the status of your Git repository.  
  
Solution:  
 git status

## Git Add

Exercise: Add a file to the staging area.  
  
Solution:  
 echo "Hello, Git!" > file.txt  
 git add file.txt  
 git status

## Git Commit

Exercise: Commit changes with a message.  
  
Solution:  
 git commit -m "Initial commit"

## Git Clone

Exercise: Clone a repository from GitHub.  
  
Solution:  
 git clone https://github.com/user/repo.git

## Git Push

Exercise: Push local changes to a remote repository.  
  
Solution:  
1. Connect your repository:  
 git remote add origin https://github.com/user/repo.git  
  
2. Push changes:  
 git push -u origin main

## Git Pull

Exercise: Pull changes from a remote repository.  
  
Solution:  
 git pull origin main

## Git Branch

Exercise: Create and switch to a new branch.  
  
Solution:  
 git branch feature-branch  
 git checkout feature-branch

## Git Merge

Exercise: Merge a branch into the main branch.  
  
Solution:  
1. Switch to the main branch:  
 git checkout main  
  
2. Merge:  
 git merge feature-branch

## Resolve Merge Conflicts

Exercise: Simulate and resolve a merge conflict.  
  
Solution:  
1. Make conflicting changes on two branches.  
2. Merge and resolve conflicts using a text editor.  
3. Add and commit resolved files:  
 git add .  
 git commit -m "Resolved conflicts"

## Git Revert

Exercise: Revert a previous commit.  
  
Solution:  
 git revert <commit-hash>

## Git Log

Exercise: View the commit history.  
  
Solution:  
 git log

## Git Ignore

Exercise: Create a .gitignore file to exclude files from tracking.  
  
Solution:  
1. Create a .gitignore file:  
 echo "node\_modules/" > .gitignore  
 git add .gitignore  
 git commit -m "Add .gitignore"

## Fork and Pull Request

Exercise: Fork a repository and create a pull request.  
  
Solution:  
1. Fork a repository on GitHub.  
2. Clone your forked repository:  
 git clone <forked-repo-URL>  
  
3. Make changes, push, and create a pull request via GitHub.

## GitLab CI/CD

Exercise: Configure a GitLab CI/CD pipeline.  
  
Solution:  
1. Create a .gitlab-ci.yml file:  
 stages:  
 - test  
  
 test\_job:  
 stage: test  
 script:  
 - echo "Running tests..."  
  
2. Push the file to trigger the pipeline.