

# Git Tutorial for Version Control

Julia Chun

jchun2@oxy.edu

Occidental College

## 1 Introduction

This document serves as a tutorial for beginners who are new to version control and git. In this tutorial, basic commands of Git are covered such as In-iting/Cloning, Adding, Committing, and Pushing. In addition, stashing, branching and merging will also be discussed in this tutorial.

## 2 Git, Github

Git is version control system used by developers to manage and collaborate efficiently on projects.

## 3 Init/Cloning

To start fresh, type "git init" on command line

Cloning a repository in Git allows a local copy of a remote repository to be made. This will allow a duplicate of the entire project history to be made on a local machine and allows changes to be made without affecting the original remote repository.

To clone a repository, the URL of the remote repository you want to clone must be located. This could be found on the main page of the repository you would like to clone. On GitHub.com, locate and click the green 'Code' button and under the tab 'Local', copy the URL. The URL could be copied in three types: HTTP, SSH, and GitHub CLI. Open Git Bash and change the current working directory to the desired location. This could be done by opening your File Navigator and right clicking on the file you would like to clone our desired repository into. Choose the option "Open GitBash here". Type "git clone" on GitBash and paste the URL.

If we wanted to clone a repository already associated with our account, we could click on 'File' and 'Clone Repository'. We select the tab 'GitHub.com' which will show all the repositories that are already forked on your account on GitHub.com so you can then take a copy of that repository onto this local machine

## 4 Adding

Adding changes involves staging modifies files for inclusion in the next commit. "Staging" refers to the process of preparing modifications for a commit. When you make modifications to files in your project, Git allows you to selectively choose which changes you would like to add in the next commit. The add command is used before the execution of the commit command.

To use the add command. Type git "add<path>" to stage a specific directory or file. You may also type "git add ." if staging all files in the entire repository is desired or git add-p to interactively stage hunks of changes.

## 5 Committing

The git commit command allows saving of a project's currently staged changes. To commit a change, type "git commit" which will launch a text editor where a summary of the changes that were committed could be written. After entering the message, save the file and close the editor to create the commit. The command 'git log' could be used to view the commit history and verify that the commit was successful. This command will display a list of commits showing details such as the message, author, and date.

To update a commit.

## 6 Pushing

Pushing changes sends commits made in your local repository to a remote repository such as GitHub.com. This will allow collaboration with others while keeping the remote repository updated with local changes.

After your changes have been committed, you may push them onto the remote repository using the 'git push' command followed by the name of the remote repository and branch that you want to be pushed. To verify that the changes have been successfully uploaded, check the remote branch on the hosting platform.

## 7 Pulling

Pulling changes allows changes to be retrieved from a remote repository and integrating them into your local repository.

To pull changes, use the 'git pull' command to retrieve changes from the remote repository and merge them to your current branch. First type 'git pull' then type the name of the remote repository and the name of the branch changes are to be pulled from.

## 8 Stashes

Stashing is a feature which allows changes to be temporarily stored that are not ready to be committed yet. This could be useful when you may need to switch to a different task or branch, but are not ready to commit changes. This allows current changes to be saved while you are able to work on something different.

Before stashing, ensure working directory has no uncommitted changes. Then, use 'git stash' command on bash which may be used when you have changes to stash but have not committed. Once your changes are stashed, you may continue working on other tasks. To revisit your stashed changes, type the 'git stash apply' command on GitBash, which will apply the changes from the most recent stash to your working directory and keep the stashed changes in the stash stack. The command 'git stash pop' could be used to apply the changes from the most recent stash to your working directory and remove the stash from the stash stack. To discard stashed changes that are no longer needed, you may remove them using the 'git stash drop' command which will remove the most recent stash from the stash stack.

## 9 Branches

Branches allow you to diverge from the main line of development and work on varying features of the project locally. The main branch in Git, named 'main' represents the primary line of development and usually contains the latest version of the project's code. Each branch in a Git repository can represent an independent line of development with its own commits. This allows effective organization and management of your project.

To create a new branch from an existing branch (usually the main branch) type 'git checkout -b' and enter the name of the new branch you would like to create between 'i' and 'c'. For instance, 'git checkout -b i\_name of branch c'. To switch branches, type 'git checkout' which will allow you to switch from working in your working directory to a different branch.

## 10 Merging

Merging allows integration of changes from one branch into another. This is typically done on a feature branch into the main branch which allows incorporation of changes into the main line of development.

To merge a branch into another branch, the target branch where changes are to be incorporated must be switched. To merge a feature branch into the main branch, switch to the main branch by typing 'git checkout main' on GitBash. Once on the target branch, type 'git merge' followed by the name of the branch to be merged on GitBash in the format 'git merge i\_branch-name c'.