
Git Assignment 2

1. What does the command 'git status' do?

The git status command is used to display the state of the repository and staging area. It allows us to see the tracked, untracked files and changes. This command will not show any commit records or information. Mostly, it is used to display the state between Git Add and Git commit command. We can check whether the changes and files are tracked or not.

2. How to delete a Git local branch?

The command is

`$ git branch -d <local-branch>`

For remote branch deletion

`$ git push origin --delete <remote-branch-name>`

3. How can I add a project to Git that already exists?

1. first create a git repository.
 2. second open git bash in existing or uploading project.
 3. perform git init
 4. git add .
 5. git commit -m "print message"
 6. git remote add github<repository url>
 7. git remote -v
 8. git push github master
-

Or

`git push origin master`

4. What exactly is a Git fork? What are the distinctions between a fork, a branch, and a clone?

A GitHub fork is a copy of a repository (repo) that sits in our account rather than the account from which we forked the data from. Once we have forked a repo, we own our forked copy. This means that we can edit the contents of our forked repository without impacting the parent repo.

A clone is a copy of all the code on the master branch. It is an exact replica of the code on github.

A branch is a slightly changed or modified section of code that meets different objectives.

Branches are not copies of each other but have same origin

Forks are local instantiations that let you make changes to someone else's codebase. They clone the repository under your profile such that it is linked and changes made to the fork can be compared or merged with the original repository over time.

5. What is the difference between HEAD, working tree, and index in GIT?

Your working tree is what is actually in the files that you are currently working on.

HEAD is a pointer to the branch or commit that you last checked out, and which will be the parent of a new commit if you make it. For instance, if you're on the master branch, then HEAD will point to master, and when you commit, that new commit will be a descendent of the revision that master pointed to, and master will be updated to point to the new commit.

The index is a staging area where the new commit is prepared. Essentially, the contents of the index are what will go into the new commit (though if you do `git commit -a`, this will automatically add all changes to files that Git knows about to the index before committing, so it will commit the current contents of your working tree). `git add` will add or update files from the working tree into your index.

6. Which GIT command is used to change branches?

The easiest way to switch branch on Git is to use the “`git checkout`” command and specify the name of the branch you want to switch to.

A quick way of switching branch on Git is to use the “`git switch`” command and specify the name of the branch you want to switch to.

7. What is the difference between GitHub and Git?

Git is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

8. What are some of the advantages of using the Variation Control System? Which programming language is used in Git?

Benefits of version control:

- Managing and Protecting the Source Code
- Keeping Track of All the Modifications Made to the Code
- Comparing Earlier Versions of the Code

*Almost all the popular programming languages are used in making the git. The majority programming language is 'c' Programming language.

9. List out some Git repository features.

Easy Project Management.

Increased Safety With Packages.

Effective Team Management.

Improved Code Writing.

Increased Code Safety.

Easy Code Hosting.
