Basic Guide: <http://rogerdudler.github.io/git-guide/>

**1)      What is GIT?**

* Distributed architecture
* Does not rely on a central server, every developer has its own copy of the repository
* Usually there is a central cloud repository where every change is gets pushed and it’s called *remote repository*

**2)      What is a repository in GIT?**

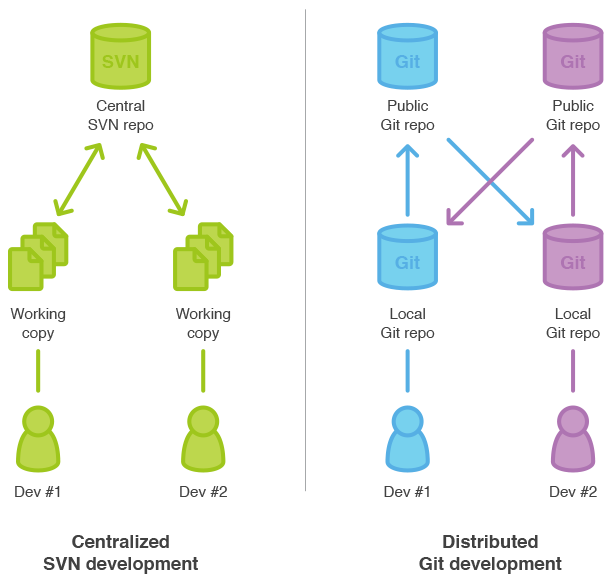
A repository contains a directory named .git, where git keeps all of its metadata for the repository. The content of the .git directory are private to git.

**3)      What is the command you can use to write a commit message?**

***git commit***

* **git commit –a:** commits the new content of all tracked files
* **git commit –m:** commits only files in staged mode

**4)      What is the difference between GIT and SVN?**

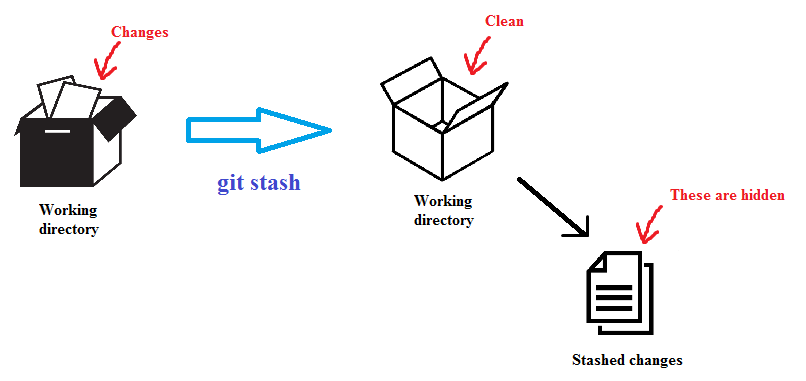


**5)      What is “Staging Area” or “Index” in GIT?**

Before pushing commits to the remote repo, it can be formatted and reviewed in an intermediate area known as ‘Staging Area’ or ‘Index’.

**6)   What is GIT stash?**

GIT stash takes the current state of the working directory and index and puts in on the stack for later and gives you back a clean working directory.  So in case if you are in the middle of something and need to jump over to the other job, and at the same time you don’t want to lose your current edits then you can use GIT stash.



**7)   What is GIT stash drop?**

Deletes the stashed changes

**8) What is the difference between git pull and git fetch?**

Git pull command pulls new changes or commits from a particular branch from your central repository and updates your target branch in your local repository.

Git fetch is also used for the same purpose but it works in a slightly different way. When you perform a git fetch, it pulls all new commits from the desired branch and stores it in a new branch in your local repository. If you want to reflect these changes in your target branch, git fetch must be followed with a git merge. Your target branch will only be updated after merging the target branch and fetched branch. Just to make it easy for you, remember the equation below:

**Git pull = git fetch + git merge**

**9)   What is the function of git clone?**

The git clone command creates a copy of an existing Git repository.  To get the copy of a central repository, ‘cloning’ is the most common way used by programmers.

**10)   What is the function of ‘git config’?**

The ‘git config’ command is a convenient way to set configuration options for your Git installation.  Behaviour of a repository, user info, preferences etc. can be defined through this command.

**11)   How can you create a repository in Git?**

**git init**. By running this command .git directory will be created in the project directory, the directory does not need to be empty.

**12)   What is ‘head’ in git and how many heads can be created in a repository?**

A ‘head’ is simply a reference to a commit object. In every repository, there is a default head referred as “Master”.  A repository can contain any number of heads.

**13)   What is the purpose of branching in GIT?**

The purpose of branching in GIT is that you can create your own branch and jump between those branches. It will allow you to go to your previous work keeping your recent work intact.

**14)   What is a ‘conflict’ in git?**

A ‘conflict’ arises when the commit that has to be merged has some change in one place, and the current commit also has a change at the same place. Git will not be able to predict which change should take precedence.

**15)   How can conflict in git resolved?**

To resolve the conflict in git, edit the files to fix the conflicting changes and then add the resolved files by running “git add” after that to commit the repaired merge,  run “git commit”.  Git remembers that you are in the middle of a merger, so it sets the parents of the commit correctly.

**16)   To delete a branch what is the command that is used?**

**git branch –d <your branch>**

**17)   What is the function of ‘git diff’ in git?**

‘git diff ’ shows the changes between commits, commit and working tree etc.

**18)   What is ‘git status’ is used for?**

As ‘Git Status’ shows you the difference between the working directory and the index, it is helpful in understanding a git more comprehensively.

**19)   What is the function of ‘git checkout’ in git?**

Switch branches or restore working tree files

**20)   What is the function of ‘git rm’?**

To remove the file from the staging area and also off your disk

**21)   What is the use of ‘git log’?**

To find specific commits in your project history- by author, date, content or history ‘git log’ is used.

**22)   What is ‘git add’ is used for?**

‘git add’ adds file changes in your existing directory to your index.

**23)   What is the function of ‘git reset’?**

The function of ‘Git Reset’ is to reset your index as well as the working directory to the state of your last commit.

**24)   What does ‘hooks’ consist of in git?**

This directory consists of Shell scripts which are activated after running the corresponding Git commands.  For example, git will try to execute the post-commit script after you run a commit.

**25)   How can you fix a broken commit?**

**git commit -amend**. By running this command, you can fix the broken commit message in the editor.

**26)   What is ‘bare repository’ in GIT?**

A “bare” repository in Git just contains the version control information and no working files (no tree) and it doesn’t contain the special .git sub-directory. Instead, it contains all the contents of the .git sub-directory directly in the main directory itself, where as working directory consist of:

1. A .git subdirectory with all the Git related revision history of your repo.
2. A working tree, or checked out copies of your project files.