**SVN –** SVN is subversion i.e repository and source code management tool. Mainly it is used for store the project and related data into the same place which is known as SVN

**SVN Merge**

**Branching**

**Tags/branch/trunck difference**

**GIT – Setup**

**All git commands**

**Git and SVN difference**

**Git Bash commands:**

**Installation and initial setup:**

1. Download latest version of Git for appropriate OS .
2. Install git by fallowing **Next** and **Finish.**
3. Open cmd or Git bash (if you elected Git Bash while installation)
4. To verify git installation run command

$ git --version

Output will be -- git version 2.14.3.windows.1

**Start git at new folder first time:**

Run the command to configure Git username and email with the help of following command

**$ git init** == to initiate git first time which create .git folder which is hidden.

**$ git config --global user.name “MayaAmol”**

**$ git config --global user.email** [**mayabcs2010@gmail.com**](mailto:mayabcs2010@gmail.com)

**Commands**

1. Create new repository

**$ git init**

1. Checkout repository

Create working copy of local repository

**$ git clone URL of repo**

e.g. **$ git clone https://github.com/MayaAmol/Devops-QA.git**

When using remote server

**$ git clone username@host:URL of repo**

1. Workflow

Local repository consist of three “trees” maintained by git

1. Working directory – holds the actual filses

**$ git add <filename>**

--- add file to the Index area

--- first step in the basic git work flow.

1. Index – which acts as staging area

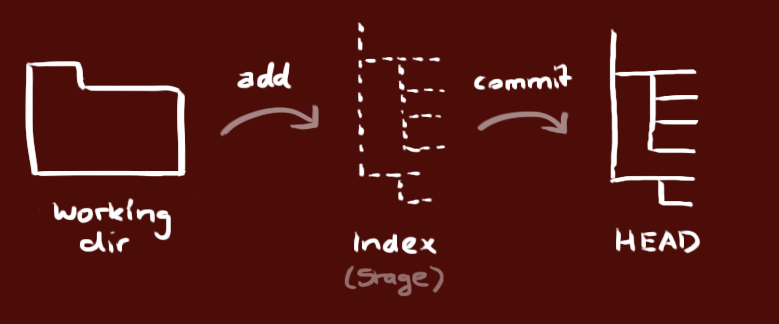
**$ git status**

--- To see the status of added and uncommitted files.

1. HEAD – points to last commit you have

**$ git commit –m “Commit Message”**

--- Now the file is committed to the HEAD, but not in your remote repository yet.



1. Push Change

Your changes now in HEAD of your local working copy. To send those changes to your remote repository execute

**Git push origin master**

Change master branch to whatever branch you want to push your changes to.

If you have not cloned an existing changes and want to connect your repository to remote server , you need to add it with

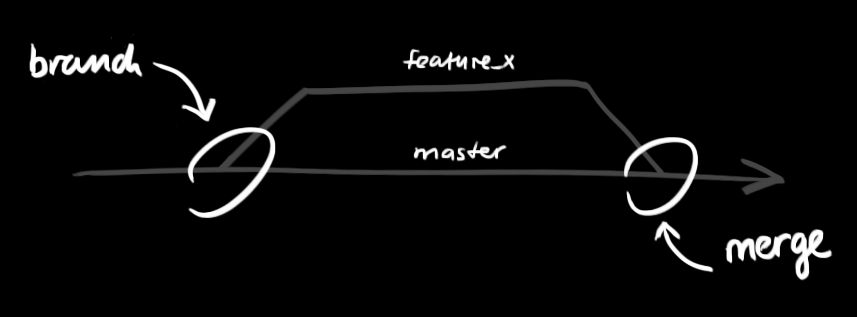
**$ git remote add origin <server>**

Now you are able to push your changes to selected remote server

1. Branching

Branches are used to develop features isolated from each other .

The master branch is the default branch when you create a repository. Use othe branches for development and merge them back to the master branch upon completion.



Create a new branch named “Branch1” and switch to it using command

**$ git checkout –b Branch1**

Switch back to master

**Git checkout master**

And delete branch again

**$ git branch –d branch1**

A branch is not available to others unless you push the branch to your remote repository.

**$ git push origin Branch1**

1. Update and merge

To update your local repository to the newest commit, run command

**$ git pull**

in your working dir to fetch and merge remote changes.

To merge another branch into your active branch (e.g. master) use

**$ git merge <branch>**

In both cases git tries to auto merge changes. Unfortunitly, this is not always possible and result in conflict. You are responsible to merge those conflict manually by editing file shown by git. After changing , you need to mark them as merged with

**$ git add <filename>**

Before merging changes, you can also preview them by using

**$ git diff <source-branch> <target-branch>**

1. Tagging

To update

1. Log
2. To update
3. Replace local changes
4. To update
5. Useful Hints
6. To update