**Git Fundamentals**

# Common commands:

|  |  |
| --- | --- |
| Git Command | Description |
| git init | Initializes a directory with git |
| git add <Filename> | Adds a file to the staging |
| git add –A | Adds all files to the staging (includes updates) |
| git commit –m “Commit message” | Commits the changes in staging to the repo |
| git status | Provides the current status |
| git log | Shows all the commits |
| git log --oneline | Shows all the commits with only comments |
| git log --graph --oneline --all --decorate | Shows all the commits with graph of all branches and and tags |
| git show head | Shows the changes in the latest commit (i.e. head) |
| git show head~1 | Shows the changes in the commit 1 before head |
| git diff head~1..head | Shows the differences between last commit and current commit |
| git tag | Shows all the tags |
| git tag “Tag Name” | Creates a tag |
| git remote | Shows all the remotes |
| git remote add <RemoteName> “<Url>”  > git remote add origin “https://github/acidx/repot.git” | Creates a remote |
| git push –u <RemoteName> <BranchName>  > git push –u origin master | Pushes the local repo to the remote repo and linked the remote to the branch so in the future all you have to do “git push”. |
| git push | Pushes local repo to remote. Assuming you have linked the remote to the master branch. |
| git fetch | Pulls the code from the remote |
| git merge <RemoteName/Branch> <LocalBranch>  > git merge origin/master master | Merges the remote repo with local repo |
| git pull | Does a pull and merge |
| git clone “<Url>”  > git clone “https://github/acidx/repo.git” | Downloads remote repository to current directory and initializes for git (you don’t have to run “git init”). |
| git branch <BranchName> | Creates a new branch (i.e. for new features) |
| git branch –d <BranchName> | Deletes a branch |
| git checkout <BranchName> | Switch to the new branch |
| git merge <BranchName> | Merges the <BranchName> into the current branch |
|  |  |

# Create New Repository

1. In your local folder, go to the root of your code base and initialize it for git.
   1. > git init
2. Add your files to the git repo
   1. > git add –A
3. Commit your changes
   1. > git commit –m “Some message”
4. Create a repository on GitHub.
5. Create a remote to your repository on GitHub
   1. > git remote add origin <https://github.com/AcidX/repo.git>
6. Push your changes the remote
   1. > git push –u origin master

# Committing Changes and Pushing To Remote Repository

1. Once your changes are ready, stage them.
   1. > git add <FileName>

or

* 1. > git add -A

1. Once you have staged all of your changes, commit them to the repo.
   1. > git commit –m “Some message”
2. Now push your changes to the remote.
   1. > git push
3. If there is a conflict, you will need to fetch from the remote and merge changes first. There are two ways to do this.
   1. Method 1:
      1. > git fetch
      2. > git merge
   2. Method 2:
      1. > git pull (git pull will do a fetch and merge)
4. If there are no merge conflicts, git will do a “Fast Forward Merge”, meaning it will update the Head to the latest commit. If there are merge conflicts, you will need to resolve them using a merge tool. See “Setting Up Merge Tool” for configuring this.
   1. > git mergetool

Once you have completed merging, save and exit the merge tool.

1. Now push your changes to the remote.
   1. > git push

# Setting up a Merge Tool

1. Install the latest version of KDiff3. KDiff3 is a 3-way merge tool and is the most popular on Unix platforms.
2. Open the Git Bash and run the following commands:
   1. > git config --global merge.tool kdiff3
   2. > git config --global mergetool.kdiff3.cmd '"C:\\Program Files\\KDiff3\\kdiff3" $BASE $LOCAL $REMOTE -o $MERGED

Note: Your path in the above command might be different.

1. Next time when you need to merge simply use:
   1. > git mergetool