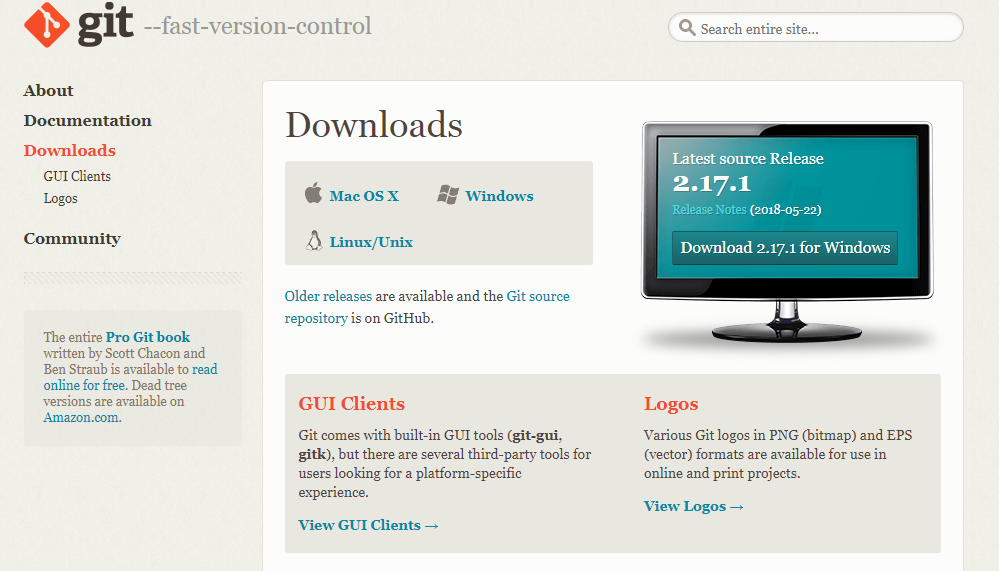
**Installation of Git:**

* Open <https://git-scm.com/downloads>.
* Select your operating system.

****

* Then your download starts and in few minutes Git will be downloaded.
* Now run the downloaded file as administrator and install the Git.

**Commands used in Git:**

Setup and Configuration commands:

* git config – This command is used to get and set repository or global options
* git help – This command is used to display help information about git

Getting and Creating Project commands:

* git init – This command is used to initialize the Git repository. This command will create a hidden folder in the project that is necessary for version control
* git clone – This command is used to creating for copying a remote project into our system.

Basic Snapshotting commands:

* git add – This command is used to add a change in the working directory to the staging area. It says that you want to include updates to a particular file in the next commit. But this doesn’t affect the repository until you commit
* git commit – This command is used to save all the work that is done until that period of time. It just takes the screenshot of the work.
* git status – This command is used to get the status of the git that may be new files, untracked files, modified files or staged.
* git-diff - This command is used to show changes between commits, commit and working tree, etc
* git-reset - This command is used to reset current HEAD to the specified state
* git-rm - This command is used to remove files from the working tree and from the index.
* git-mv - This command is used to move or rename a file, a directory, or a symlink

Branching and Merging commands:

* git branch – This command is used to show the branched that are being worked in our system.
* git-checkout - This command is used to switch branches or restore working tree files
* git-merge - This command is used to join two or more development histories together.
* git-mergetool - This command is used to run merge conflict resolution tools to resolve merge conflicts.
* git-log - This command is used to show commit logs.
* git-stash - This command is used to stash the changes in a dirty working directory away.
* git-tag - This command is used to create, list, delete or verify a tag object signed with GPG.
* git-worktree - This command is used to manage multiple working trees.

Sharing and Updating Projects commands:

* git-fetch - This command is used to download objects and refs from another repository.
* git-pull - This command is used to fetch from and integrate with another repository or a local branch
* git-push - This command is used to update remote refs along with associated objects.
* git-remote - This command is used to manage set of tracked repositories.
* git-submodule - This command is used to initialize, update or inspect submodules

Inspection and Comparison commands:

* git-show - This command is used to show various types of objects.
* git-diff - This command is used to show changes between commits, commit and working tree, etc.
* git-shortlog - This command is used to summarize git log output.
* git-describe - This command is used to give an object a human readable name based on an available ref

Patching commands:

* git-apply - This command is used to apply a patch to files and/or to the index.
* git-cherry-pick - This command is used to apply the changes introduced by some existing commits.
* git-rebase - This command is used to reapply commits on top of another base tip.
* git-revert - This command is used to reverts some existing commits.

Debugging commands:

* git-bisect - This command is used to uses binary search to find the commit that introduced a bug.
* git-blame - This command is used to shows what revision and author last modified each line of a file.
* git-grep – This command is used to prints the lines matching a pattern

Email commands:

* git-am - This command is used to apply a series of patches from a mailbox.
* git-format-patch - This command is used to prepare patches for e-mail submission.
* git-send-email - This command is used to send a collection of patches as emails.
* git-request-pull - This command is used to generate a summary of pending changes

External Systems commands:

* git-svn – This command is used to provide bidirectional operation between a Subversion repository and Git.
* git-fast-import – This command is used to give backend for fast Git data importers.

**Creating new Repository:**

There are ways to create a repository. Here are list of ways and the steps that should be followed to create a new repository.

A new repository from scratch:

These are steps that should be followed when we create a repository from a scratch-

* Create a directory to contain the project.
* Go into the new directory.
* Type git init.
* Write some code.
* Type git add to add the files (see the typical use page).
* Type git commit.

A new repo from an existing project:

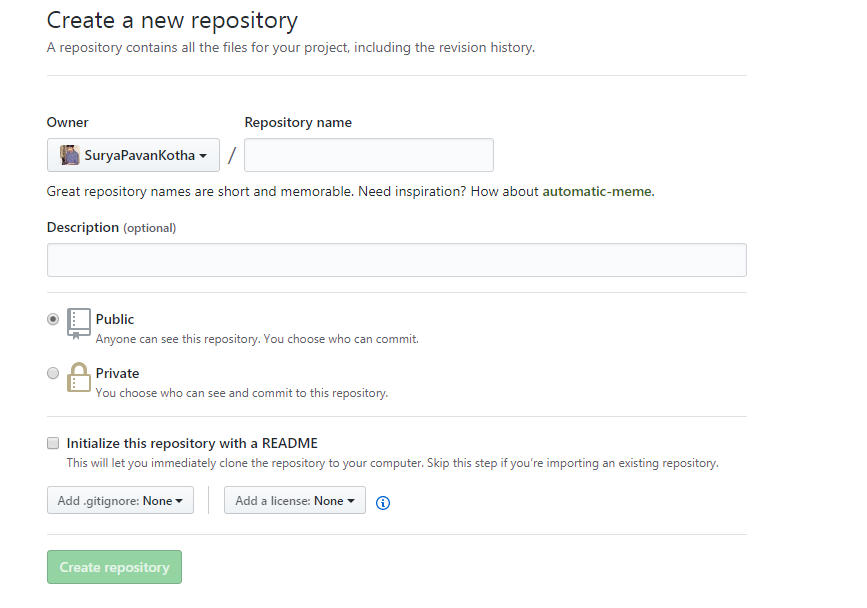
These are steps to be followed when we want to create a A new repo from an existing project

* Go into the directory containing the project.
* Type git init.
* Type git add to add all of the relevant files.
* You’ll probably want to create a .gitignore file right away, to indicate all of the files you don’t want to track. Use git add .gitignore, too.
* Type git commit

Now after the creation of repository, now we can share it with others by using Github. For this we have to follow the below steps-

* Go to github.
* Log in to your account.
* Click the new repository button in the top-right. You’ll have an option there to initialize the repository with a README file, but I don’t.
* Click the “Create repository” button.

By looking below image we can get an idea how our Github account resembles



Now, follow the second set of instructions, to “Push an existing repository…”

$ git remote add origin git@github.com:username/new\_repo

$ git push -u origin master

Actually, the first line of the instructions will say

$ git remote add origin <https://github.com/username/new_repo>

In this way we can create a repository.