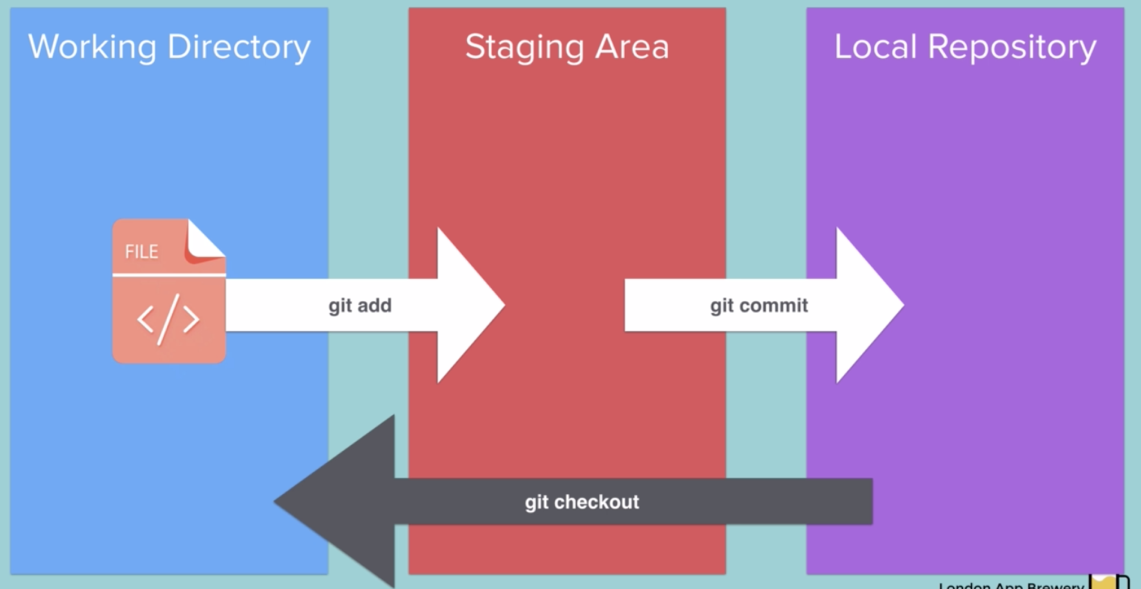
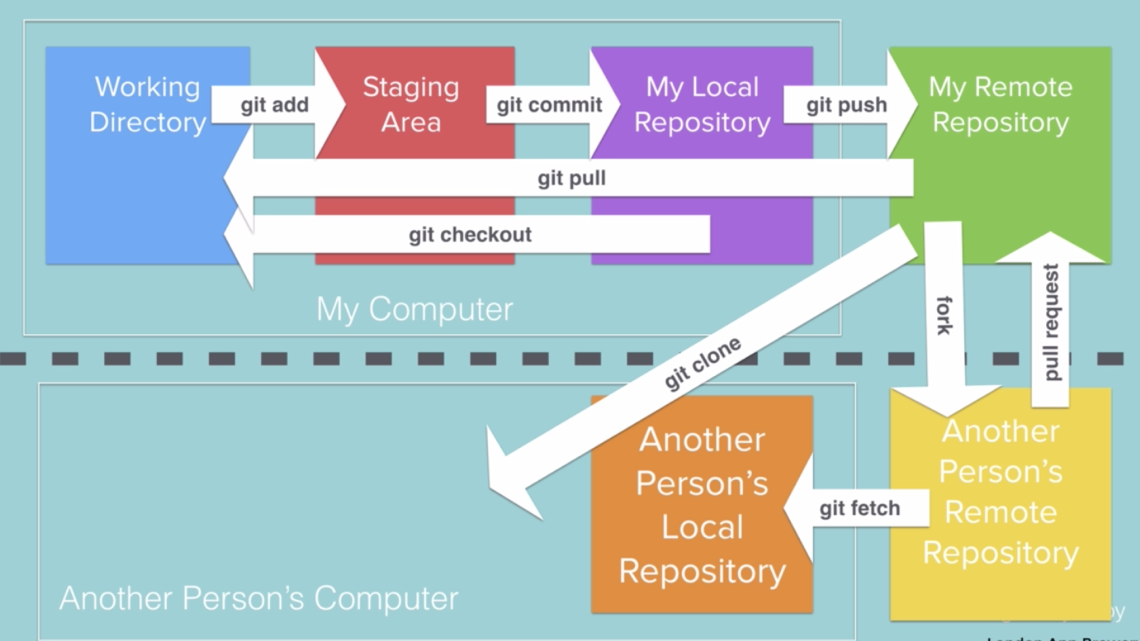
**GIT REPOSITORY**

1. **Used for version control**





git init // initialise git

git status // check status of the files which are in the staging area. A staging area is an intermediate place where you can pick and choose which files we need to commit. There will be files in red color which are untracked and these are inside working dir but not in staging area. So to add it into staging area, we use command;

git add ‘filename’ // adds a single untracked file to staging area.

git add . //adds all untracked files to staging area

git commit -m “message” // eg: Complete chapter 1”

git log // gives a log of the commits made

git diff ‘filename’ // gives the differences in the two files

git checkout ‘filename’ // checkouts the last version that has been committed to the local repository.

1. **Remote Repository**

git remote add origin ‘url from github’ // creates a remote repository. Origin is the name of the remote repository

eg:

git remote add origin *https://github.com/2rahulsk/forkify.git*

git push -u origin master // pushes local repository to remote repository. we are pushing to the master branch.

The local repository is the .git file and the remote repository is the Github.

1. **Gitignore**

touch .gitignore // creates a hidden file which could be found using ls -a command

git rm –cached -r . // undo or remove files from staging area.

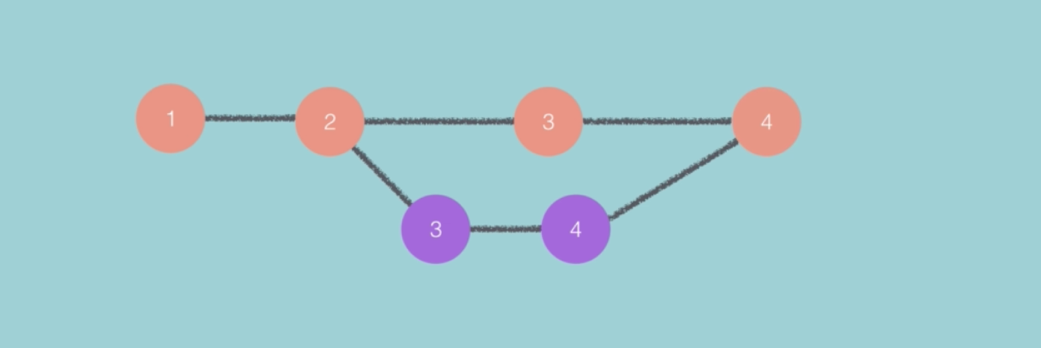
After this step we can add the filenames of files that needs to be ignored while adding and committing into the .gitignore file. We add names of file in separate lines. In gitignore file we can use # sign to use a comment. We can got to gitignore repo is github which gives a template for different programming languages.

1. **Cloning**

It’s a way to pull down all the versions of a particular remote repo and store them in a working directory.

git clone ‘url of the repo’

1. **Branching and Merging**

****

git branch ‘branch-name’ // creates a new branch

git branch // shows the branches with an \* near the current branch.

git checkout ‘branch-name’ // switches to the branch specified

follow the same steps in adding and committing to the new branch

To Merge,

git branch master // move to master branch

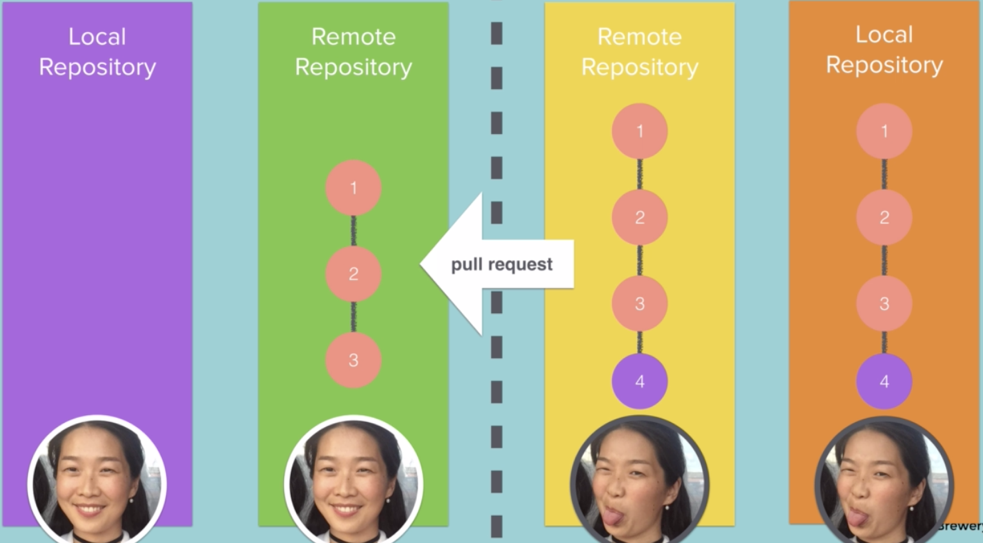
git merge ‘branch-name’ // merge

git push -u origin master

:q! – quit with save

1. **Forking and Pull requests**

Copying a remote repository and creating a workable copy for working is called as forking. People working in same team can CLONE while others who wants to use your product can FORK ( open source collaboration) the repository.



Once we fork a repo, we own it and can make changes that we want to it. Once we made some changes and push changes to remote repository and the can do a pull request to the remote repo of the original user. The user of the original remote repo can review the changes and then pull the remote repo of the other user and the merge it.

Steps:

1. Fork the original repo
2. Make changes
3. Make a pull request
4. Wait for the other person to approve the pull request and merge the changes you suggested