Git & Github Refresher

By - Shashwat Agrawal
Handbook: https://docs.github.com/en/get-started/using-git/about-git

---

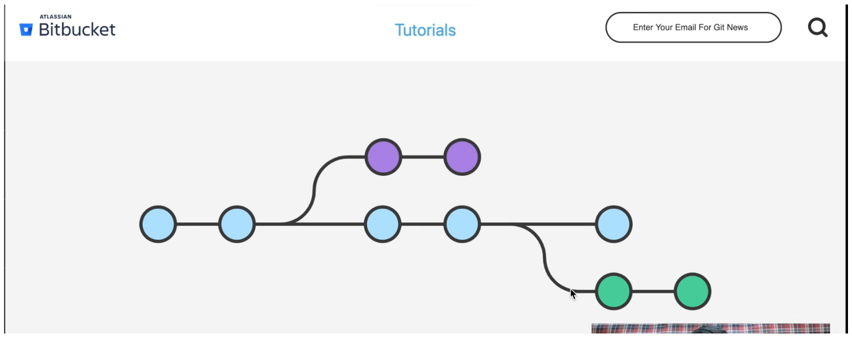
Introduction

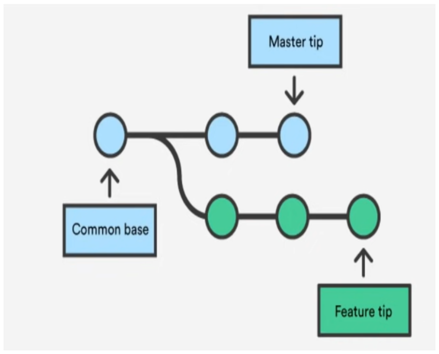
- Git is a version control system to keep a track of files.
- Who changed? What changed? When changed? Why changed? etc.

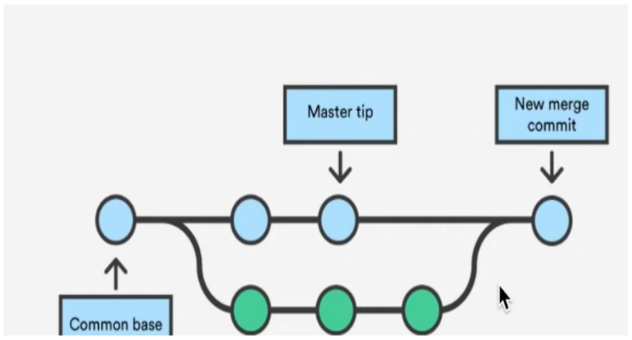
- GitHub is a cloud-based web platform that is used for your project management.
- A repository is a folder containing your project files.
- Project management in Git & GitHub is done using repositories.

- Download and install Git in the system.
- Check the version of git using
    - git --version

- A branch is a series of code changes



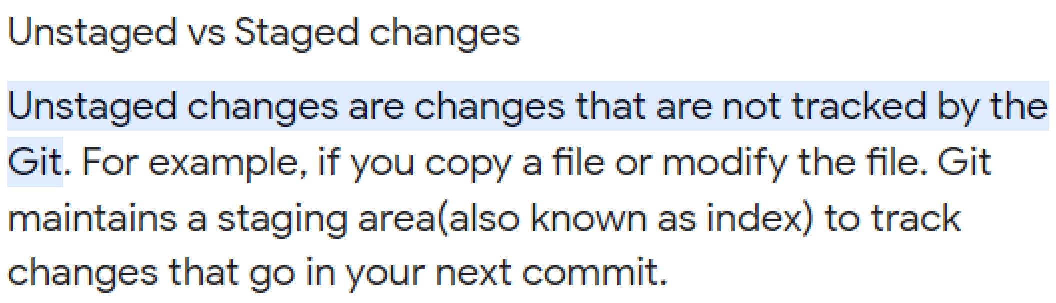




Git Commands

- **git init** – creating a new/blank repository.
    - .git folder will be created which will contain all files required to maintain a git repository.
- **git clone url** - cloning a repo from GitHub to the local machine
    - Eg, git clone https://github.com/shay-ag/Fashify.git
- **git status** - checking current repo status

- **git diff filename** - to know the changes made in the file named as *filename*
    - Eg, git diff index.html
- Commit means locking a file. To commit your file it should be in the staging area.
- Staging Area & Unstaging Area

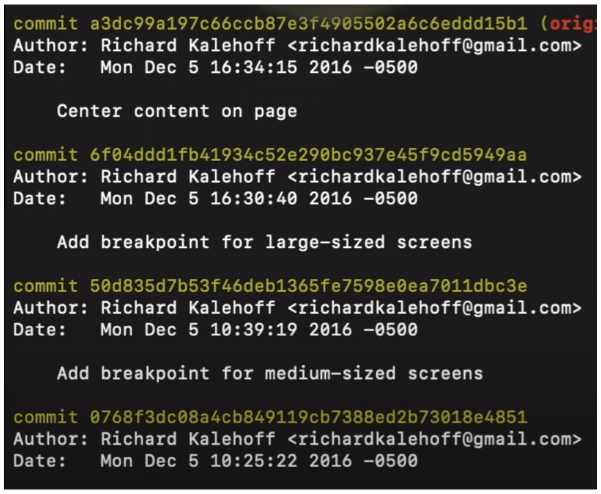


Unstaged vs Staged changes

Unstaged changes are changes that are not tracked by the Git. For example, if you copy a file or modify the file. Git maintains a staging area(also known as index) to track changes that go in your next commit.

- **git add filename** - to add the file named as the *filename* in the staging area in order to commit it.
- **git add .** - to add all files in the staged area.
- **git commit -m “message”** - to commit the file
- **git commit --amend** - to commit changes in the previous commit only. After running this command VM editor will show up. Next steps to save: “ESC w q ENTER”

- **git log** - to show all previous commits



```
commit a3dc99a197c66ccb87e3f4905502a6c6eddd15b1 (orig
Author: Richard Kalehoff <richardkalehoff@gmail.com>
Date:   Mon Dec 5 16:34:15 2016 -0500

    Center content on page

commit 6f04ddd1fb41934c52e290bc937e45f9cd5949aa
Author: Richard Kalehoff <richardkalehoff@gmail.com>
Date:   Mon Dec 5 16:30:40 2016 -0500

    Add breakpoint for large-sized screens

commit 50d835d7b53f46deb1365fe7598e0ea7011dbc3e
Author: Richard Kalehoff <richardkalehoff@gmail.com>
Date:   Mon Dec 5 10:39:19 2016 -0500

    Add breakpoint for medium-sized screens

commit 0768f3dc08a4cb849119cb7388ed2b73018e4851
Author: Richard Kalehoff <richardkalehoff@gmail.com>
Date:   Mon Dec 5 10:25:22 2016 -0500
```

- **git log -5** - this will show 5 previous git commit logs, thus you may enter the number of logs you want to view.

- **git show commitId** - to show changes in the particular commit. This particular commit is known by the commit hash value which is in yellow color number.
    - Eg, git show qwedrfghlasmcnkjsvbcncnvm
- **git branch** - showing the current branch you're on.
- **git branch new** - creating a *new* branch
- **git checkout new** - to shift to *new* branch
- **git checkout -d newbranch** - delete the branch. Remember to delete a branch you must be on master or any branch other than the branch you're deleting.
- **git merge new** - to merger branch new to branch master

  Remember: It is a good practice to merge a branch from the branch which is behind. That is if the new branch is ahead of the master branch then go to the master branch and then run the merge command

- **git restore filename -** to restore/revert the changes in that file
- **git reset --hard HEAD^** - to revert the last commit that is discard the latest commit along with the changes
- **git reset --soft HEAD^** - to revert the last commit that is discard the latest commit but changes will not be removed

- **git pull** - to sync the commit history of local and remote
- **git push** - to push the changes from the local to the remote git repository

---