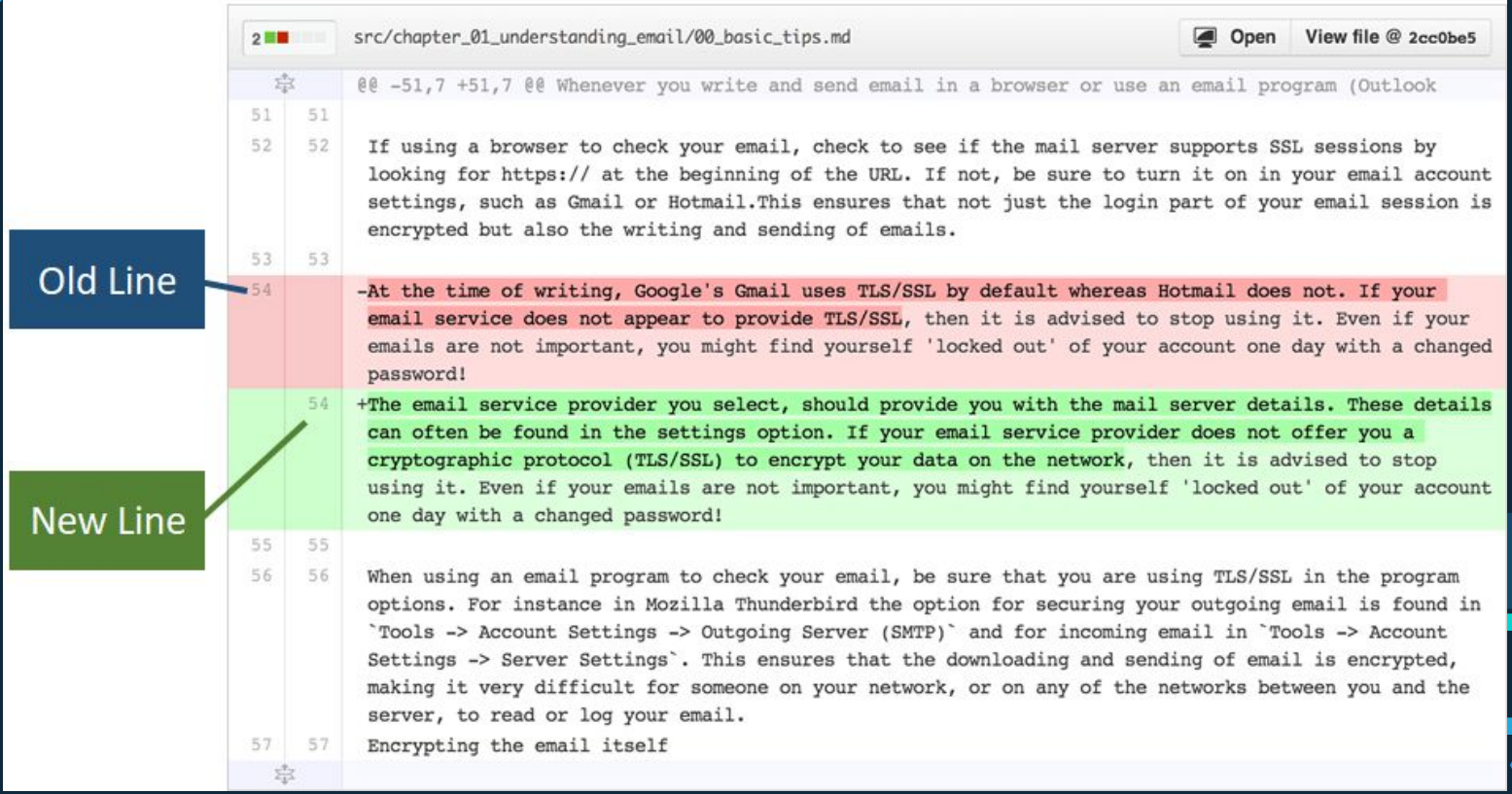
Introduction

Git is the free and open source distributed version control system that's responsible for everything GitHub related that happens locally on your computer. This cheat sheet features the most important and commonly used Git commands for easy reference. Git has a remote repository which is stored in a server and a local repository which is stored in the computer of each developer.

- Version Control System
  - Allows you to track changes in a project



Git Create and Clone Repository

- First create a repo in github using command line or in github website.
- Pull or clone the code to your desktop using git client
- Modify the code or add a new file.
- Add the file.
- Commit the file for push.
- Push the changes to github.

Create and clone repo

STEPS:

Create Repo (either at github web or at your local system ,local system should be in sync with github web

So that create repo will reflect there)

Clone repo(if your repo is created at github web)

Add or modify files

Add files

Push changes to github web or local company repo

git pull code

Download the latest changes

```
git pull REMOTE <name-of-branch>
```

View your remote repositories

To view your remote repositories, type:

```
git remote -v
```

Add a remote repository

To add a link to a remote repository:

```
git remote add <source-name> <repository-path>
```

branch merge

John creates b1

Joy creates b2

Dolly creates b3

Sita creates b4

All these branches must be merged with master to get all the changes in master.

create branch

$git pull <github link> branch name

or

pull from master

#create a new branch

$git checkout -b gaurav

branch merge

Edit the file

#Add the file

$git add .

#commit the file

$git commit

git merge

#checkout to master

#merge the branch

$git merge gaurav

$git push

file add & commit

If conflict arises then manually edit and then add the file

$git add file

$git commit -m “manually edited file”

$git push

branch merge

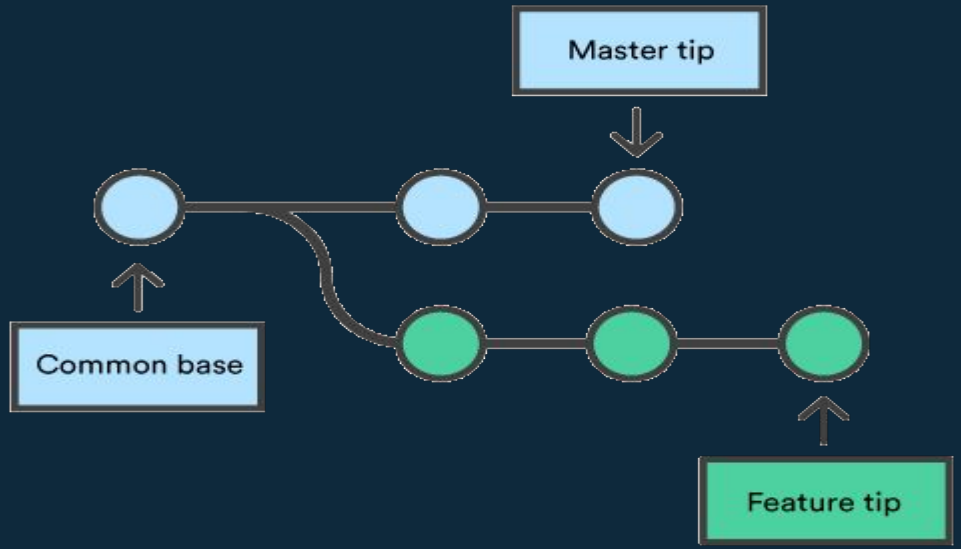
If conflict arises then manually edit and then add the file

$git add file

$git commit -m “manually edited file”

$git push

3-way branch merge

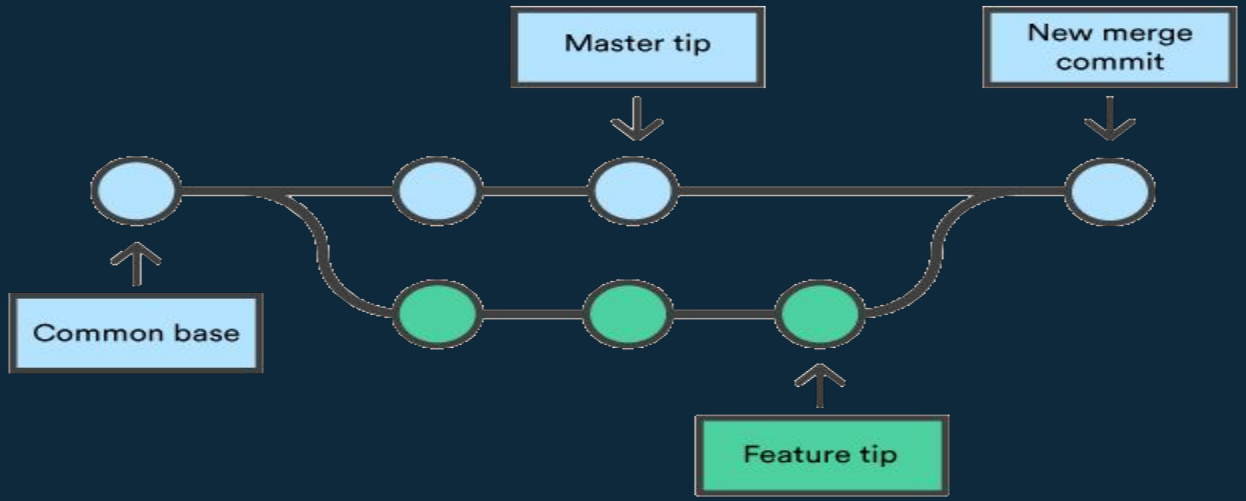


branch merge

When multiple people are working on a project ,its subtle task to manage changes with version .But git does it smartly by maintaining the branch and version details of each commit and also branch details.

We can choose specific changes ,which we want and merge it to master branch (main code) which will be used for delivery.

branch merge



merge vs rebase

local & remote branch

```
(base) apples-MacBook-Pro:SampleCode apple$ git  branch -a
  b1
  b2
  b3
  b4
* master
```

remote branch and merged to master

```
(base) apples-MacBook-Pro:SampleCode apple$ git  branch -a
  b1
  b2
  b3
  b4
* master
```

tag version

```
base) apples-MacBook-Pro:SampleCode apple$ git tag -a v2.9 -m 'version 2.9'
(base) apples-MacBook-Pro:SampleCode apple$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 158 bytes | 158.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/penguintechlinux/SampleCode.git
 * [new tag]         v2.9 -> v2.9
```

git merge

#checkout to your branch

$git checkout gaurav

Make changes

$git add file

$git commit -m “changes”

$git merge gaurav master

$git push

Branching

#checkout to your branch

$git checkout gaurav

Make changes

$git add file

$git commit -m “changes for issue of ipv4”

$git merge gaurav master

$git push

Conflict resolution

Check manually after opening the file ,do the required changes and then

git add filename

git commit -m “conflict resolved”

git push

And proceed for next steps…..

Best practices

Check whether your branch is updated or not.

Check git status.

Check git branch and local and remote branches and branches which is left out.

Avoid making changes in master branch

Provide a proper information in git commit message.

Check the difference in code with previous versions.

Merge your local branches with master when required and manage conflict wisely.

Pitfalls

- Discard local file modifications. ...
- Undo local commits. ...
- Remove a file from git without removing it from your file system. ...
- Edit a commit message. ...
- Clean up local commits before pushing. ...
- Reverting pushed commits. ...
- Avoid repeated merge conflicts. ...
- Find the commit that broke something after a merge.