

# Table of Contents

<b>Arnab's GitHub Instructions.....</b>	<b>1</b>
How to fork a repository?.....	1
How to add an existing project to GitHub using command line?.....	1
How to update/Sync a gitHub repository?.....	1
How to merge a pull request in your local/forked gitHub repository?.....	2

# Arnab's GitHub Instructions

---

Welcome!

Here you can find useful instructions for gitHub in Q&A format.

## How to fork a repository?

### How to add an existing project to GitHub using command line?

1. Create a new repository [🔗](#) on GitHub. To avoid errors, do not initialize the new repository with *README*, license, or *gitignore* files. You can add these files after your project has been pushed to GitHub.

2. Open Terminal.

3. Change the current working directory to your local project.

4. Initialize the local directory as a Git repository:

```
git init
```

5. Add the files in your new local repository. This stages them for the first commit:

```
git add .
```

6. Commit the files that you've staged in your local repository:

```
git commit -m "First commit"
```

7. At the top of your GitHub repository's Quick Setup page, click to copy the remote repository URL.

8. In Terminal, add the URL for the remote repository [🔗](#) where your local repository will be pushed:

```
git remote add origin remote repository URL
```

```
# Sets the new remote
```

```
git remote -v
```

```
# Verifies the new remote URL
```

9. Push the changes [🔗](#) in your local repository to GitHub:

```
git push origin master
```

### How to update/Sync a gitHub repository?

1. Clone your fork:

```
git clone git@githubNOSPAMPLEASE.com:YOUR-USERNAME/YOUR-FORKED-REPO.git
```

2. Add remote from original repository in your forked repository:

```
cd into/cloned/fork-repo
```

```
git remote add upstream git://github.com/ORIGINAL-DEV-USERNAME/REPO-YOU-FORKED-FROM.git
```

```
git fetch upstream
```

3. Updating your fork from original repo to keep up with their changes:

```
git pull upstream master
```

4. Push changes to your forked area:

```
git push
```

#### **How to merge a pull request in your local/forked gitHub repository?**

If you have a forked repository (which you don't own) where pull request is done by some user and it is not merged yet in the original repository. But you want to merge it in your local area. Do the following,

1. Go to the forked directory in your local computer.

2. `git pull https://github.com/[username_ who_made_the_pullrequest]/[Repo_name].git [branch_name_which_is_requested_to_merge]`

Note: You may find some conflict. It means while merging it finds some files which are already changed by you but you have not committed. Please look at the question regarding the merge conflict.

#### **How to resolve a merge conflict caused by competing line changes?**

Merge conflicts occur when competing changes are made to the same line of a file, or when one person edits a file and another person deletes the same file.

To resolve a merge conflict caused by competing line changes, you must choose which changes to incorporate from the different branches in a new commit.

For example, if you and another person both edited the file *Test.txt* on the same lines in different branches of the same Git repository, you'll get a merge conflict error when you try to merge these branches. You must resolve this merge conflict with a new commit before you can merge these branches.

1. Go to the Forked directory

2. Generate a list of the files affected by the merge conflict using:

```
"git status"
```

3. Open the file with your favorite text editor and navigate to the file that has merge conflicts.

4. To see the beginning of the merge conflict in your file, search the file for the conflict marker <<<<<<<. When you open the file in your text editor, you'll see the changes from the HEAD or base branch after the line <<<<<<< HEAD. Next, you'll see =====, which divides your changes from the changes in the other branch, followed by >>>>>>> BRANCH-NAME. For example, if one person wrote "This is my file" in the base or HEAD branch and another person wrote "This is our file" in the compare branch or the branch which you want to merge.

5. Decide if you want to keep only your branch's changes, keep only the other branch's changes, or make a brand new change, which may incorporate changes from both branches. Delete the conflict markers <<<<<<<, =====, >>>>>>> and make the changes you want in the final merge. In this example, the second change is

incorporated into the final merge: "This is our file"

6. Add or stage your changes:

```
"git add ."
```

7. Commit your changes with a comment:

```
git commit -m "your message"
```

8. You can now merge the branches on the command line or push your changes to your remote repository [on GitHub](#) and merge your changes [in a pull request](#).

## How to

-- ArnabPurohit - 2018-06-02

---

This topic: Main > ArnabsGitHubInstructions

Topic revision: r4 - 2019-09-11 - ArnabPurohit



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback