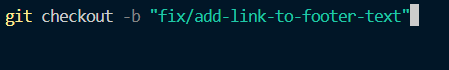
Steps to Open Source Contribution:

1. Create a new issue to the original repo. (If you wanted to contribute on the new issue)  
   Ex:  
     
     
    OR  
     
   From the **Issues** section you can pick up any issue and start working on it. For beginner, you can go with “good first issue” first by filtering out from the label.
2. Then, Fork the original Repo.   
   For best practise: you can go with the original repo name.
3. Need to clone the fork repo, in your local sys.  
   Ex:  
     
     
   Note: Since, we’ll be doing changes on our fork repo. Hence, Origin will point to our fork repo.
4. Create a new branch and switch to it.  
   Ex:  
     
     
   Make sure you’re in the newly created branch.  
   To check: git branch
5. Do your Changes (This is called Contribution).
6. “Staged” the changes and commit it.  
   Ex:  
   git add .  
   git commit -m “add link to footer text”
7. Push the commit to upstream:  
   Ex:  
   
8. Click on Compare & pull Request on your fork repo. It’ll redirect you to the main repo and need to send pull request by explaining what you’ve fixed.  
   Ex:  
     
     
   Make sure,

* fix: keyword should be there for bug fixes.  
  Or feat: for feature added.
* Added Fixes: Issue number in the description.

1. Then the, reviewer/maintainer/owner of original (upstream) repo, will review your pull request and press “Merge pull request” to add your changes to the production.

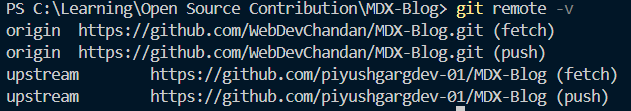
Note: if you close pull request, then maintainer will not able to merge it.

1. **Two points** need to keep in mind, before re-contributing to another issue on the same upstream (open source repo). As might be other contributor’s issue or your previous issue got merged. So need to **Sync up, Upstream Repo, Github Fork Repo, Local Fork Repo**:

* **First Switch to Main Branch of local fork repo**: Switch the previously created new branch back to the main branch first in your local rep. If we don’t do so, it might create a confliction to multiple issues in the same pull request.  
  *git switch main*
* **Update the local fork repo main branch**: Before started working on your another issue, make sure to update the local fork repo main branch with the previously merged pull request.

1. To update the local fork repo main branch with the upstream (original repo main branch):

* **Add remote of upstream (original repo)**  
  Ex:  
  



* **Fetch/Pull the Upstream Update from main branch**: To fetch the update found in the original repo, use the below command:  
  *git pull upstream main* or  
  *git fetch --all*  
    
  Note: It will update your local fork main with all the latest pull request merged in upstream found. If no update found, it’ll return Already up to date.
* **Merge “Upstream main” to “local fork repo main”**:  
  If update found, then we can go with the below command-  
    
  If no update found, and still we go with the above command, we’ll encounter with the below message:  
  

1. Now Your Upstream & Local Fork repo are sync up. But github Fork repo is not sync with the local fork repo.  
     
   Run the below command to sync Github Fork Repo:  
   *git push* or  
   *git push origin main*Now all, **Upstream (origin repo main), Local Fork Repo main & Github Fork Repo main** is **Sync up**.

**Note:**

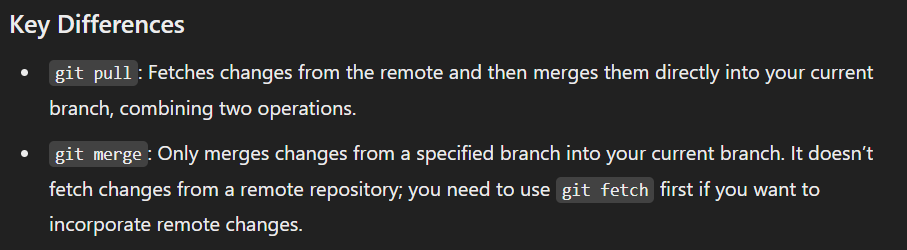
**Upstream:** Means the original Repo default (Main/Master) branch. Or the repo which you forked from.

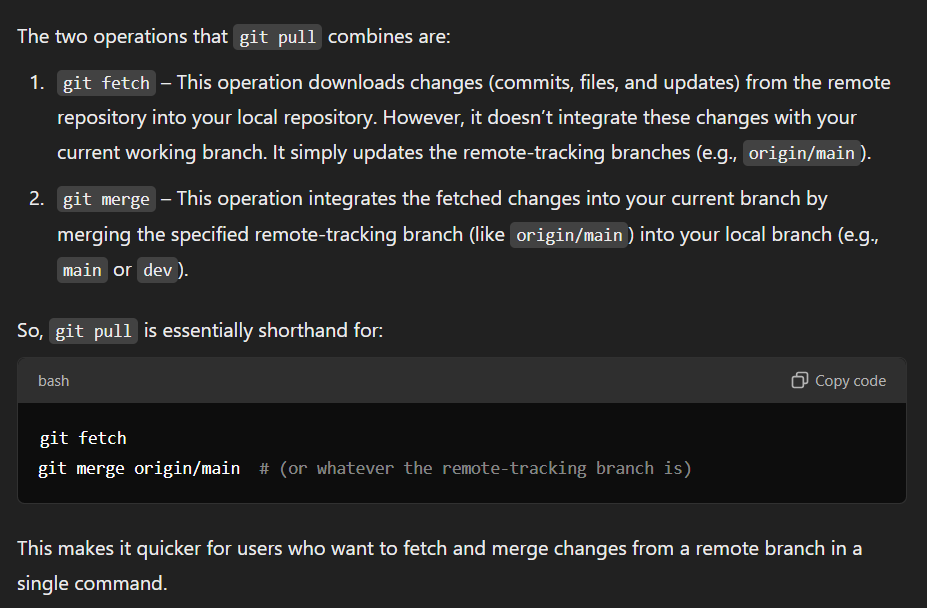
**Fork**: Means copy the original repo. We can fork a repo only once.

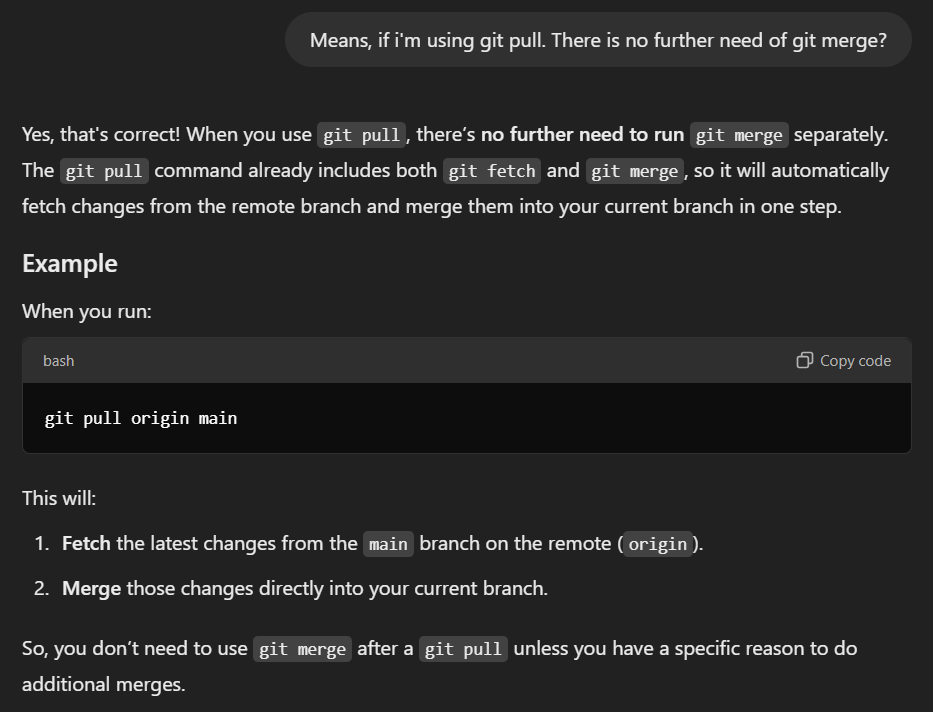
**Pull Request**: A way of requesting to add our changes or added code to the default branch (main branch) of upstream (original repo) via newly created custom branch.

**Merge Conflict:** When changes found at the same line for two or more pull request.  
let say, User-1’s pull request got merged. Now User-2’s did some changes for the same line of user-1’s pull request and now going to send a pull request for merging it, then user-2 will see a merge-conflict from github.  
  
Now user-2 need to resolve it, from vs code git bash terminal. Note: It’s a critical task, that should be taken care of very carefully.

* First *git pull upstream main*
* Then VS code give you options for git merge. Then analyse it and go with the suitable option and edit it.

**Difference between git pull & git merge:  
**

**Git Pull:  
**

****