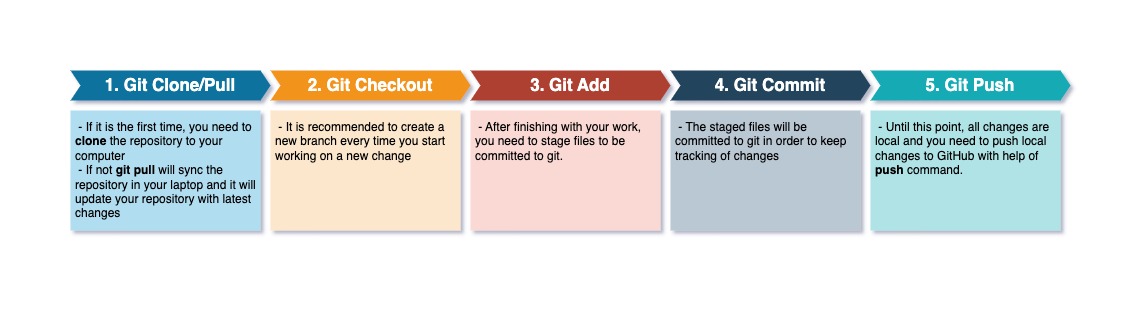
# Git lifecycle

Assume you need to work on a project where the code is in a GitHub Repository. The recommendation is changing code in your workstation and then publish changes to GitHub for review and merge. I recommend a 10 easy steps process to follow to reduce the chance of having bugs in code as well as reduce the chance of git merge conflict. I divide these steps to 2 parts: - Part 1: work on local repository - Part 2: work on GitHub UI

## Part 1: development in your workstation/computer

 I provide Git Commands for each steps in part 1 if you wish to use Linux/MacOS Terminal, Git Bash, Windows Powershell or Windows Command Prompt. If you wish to use an IDE or a Text Editor wiht github integration feature, please consult documentations from the vendor. Keep in mind **All steps are same for IDEs, text editors, terminals, powershel and command prompt**

### All git commands you need for steps 1 to 5

## Step 1  
 # if you clone repositry  
 git clone <repository-clone-url>  
 # example  
 git clone https://github.com/github/super-linter.git  
   
 # git pull  
 git pull  
   
 ## Step 2  
 # create a new branche  
 git checkout -b <new-branch-name-with-no-space>  
 example  
 git checkout -b task-123  
   
 ## Step 3  
 # staging files  
 git add <file-name>  
 # example  
 git add README.md  
   
 ## Step 4  
 # git commit  
 git commit -m "commit message"  
 # example  
 git commit -m "task 123 updated README.md"  
   
 ## Step 5  
 # git push  
 # if branch does not exist in github  
 git push -u origin branche-name  
 # if git branch does exist in github  
 git push

### Step 1 Git Clone or Git Pull

At this step you need to make sure you have latest and gratest code version in you computer. to do so follow one of below steps

 - If it is the first time, you need to clone the repository to your computer   - the command will be git clone <repository-clone-url>   - For example git clone https://github.com/github/super-linter.git  - If not git pull will sync the repository in your laptop and it will update your repository with latest changes   - Make sure you are on **default branch** then run git pull command   - If your default branch name is main just type git checkout main then git pull

### Step 2 create a new branch

* It is recommended to create a new branch every time you start working on new changes. Here are

### Step 3 stage your changes

 - After finishing with your work, you need to stage files to be committed to git.

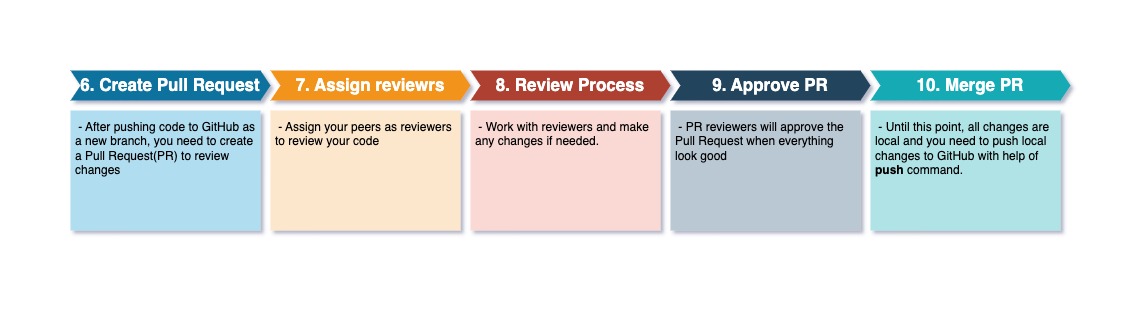
### Step 4 commit changes

 - The staged files will be committed to git in order to keep tracking of changes

### Step 5 push local changes to GitHub

 - Until this point, all changes are local and you need to push local changes to GitHub with help of push command.

## Part 2: Review and merge code in GitHub



pull-request

### Step 6 Create a Pull Request to review changes by your peers

  - After pushing code to GitHub as a new branch, you need to create a Pull Request(PR) to review changes

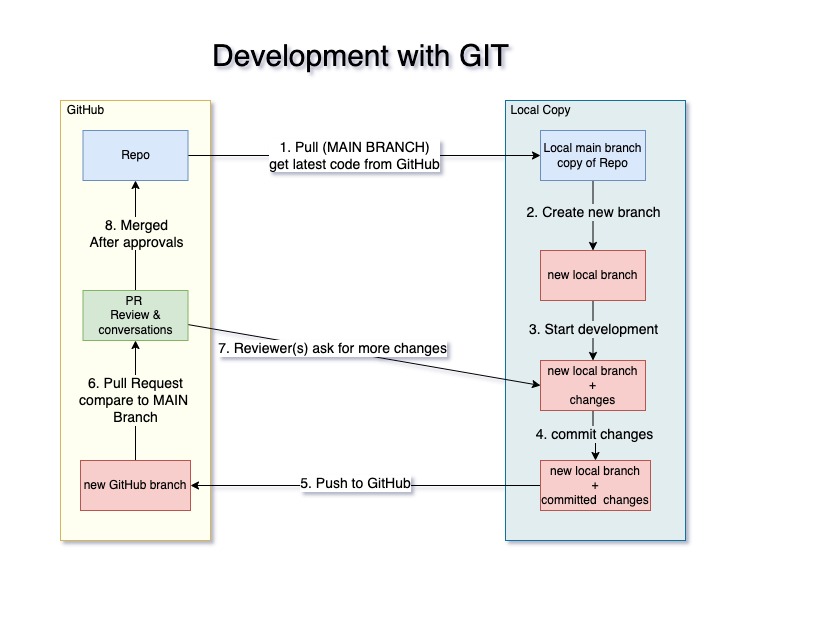
### Step 7 Assign your peers to the Pull Request

 - Assign your peers as reviewers to review your code   ### Step 8 Work with peers to get approval  - Work with reviewers and make any changes if needed.   ### Step 9 Your changes have been approved by your peers  - PR reviewers will approve the Pull Request when everything look good

### Step 10 Merge your code to main branch

 - In the final step: merge the Pull Request and changes will be available in the repository’s default branch.

## Developing code with GitHub at a glance



git-cycle

## External refs

* [Git distributed Version Control System](https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control#:~:text=Distributed%20Version%20Control,all%20the%20data.)