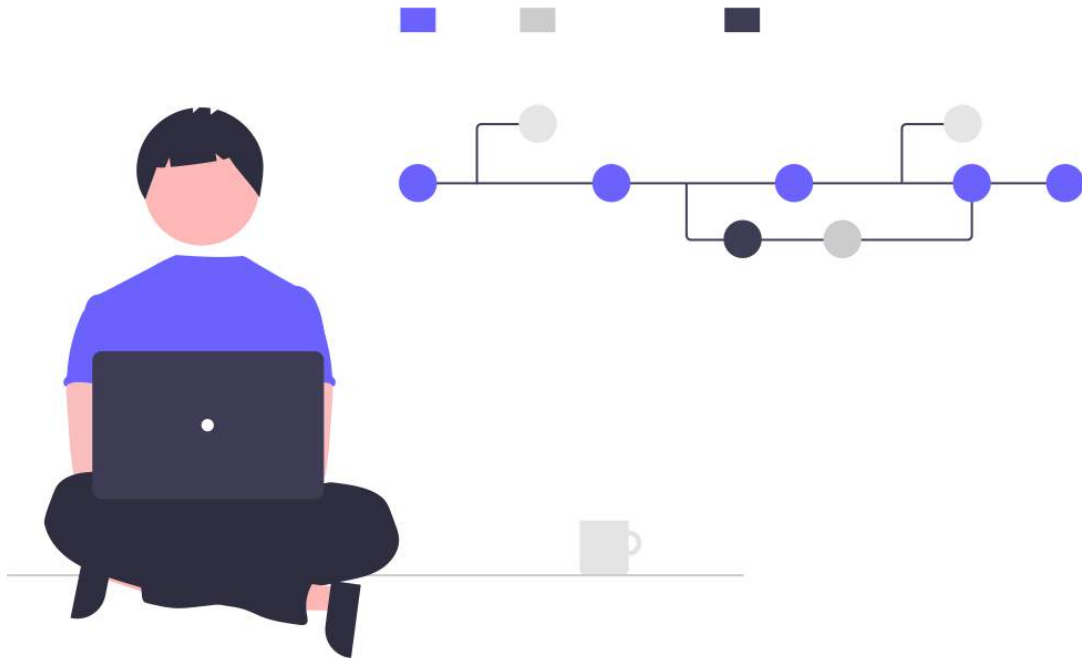# Practical **Git** Guide

Practical guide to everyday Git commands

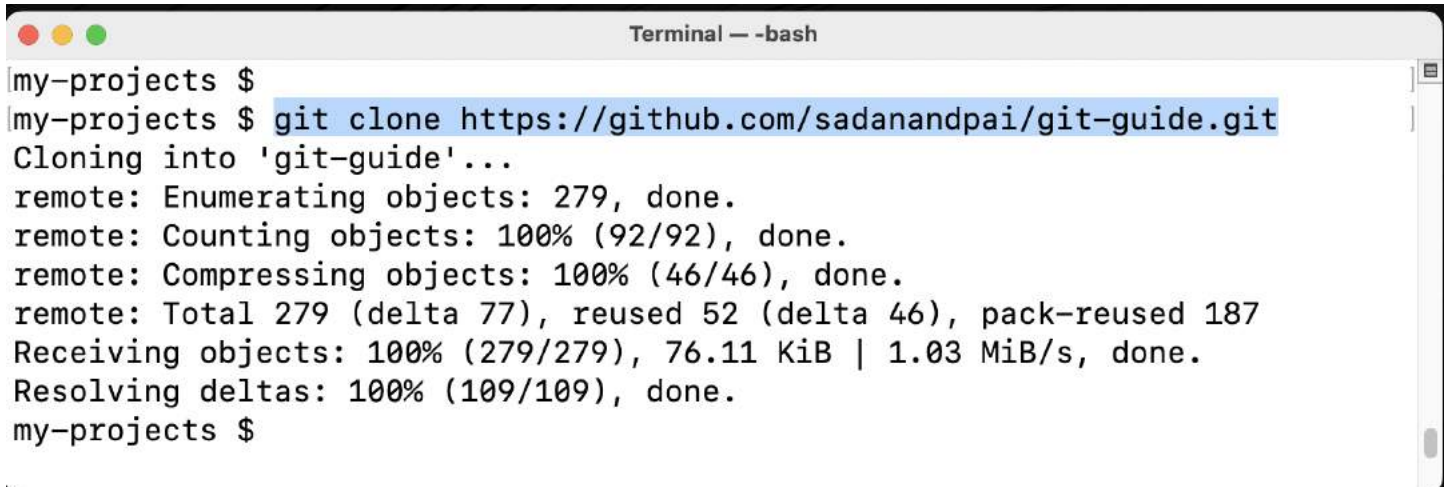Solutions to **90%** of your daily git challenges

Reference material to the below concepts with live examples

- Clone the repo
- Switch between branches
- Pulling the changes
- Committing the changes
- Pushing the changes
- Managing the stash
- Branching out
- Raising a PR

- Merge the branches
- Rebase the branches
- Resolving the conflicts
- Squash the commits
- Revert and Reset
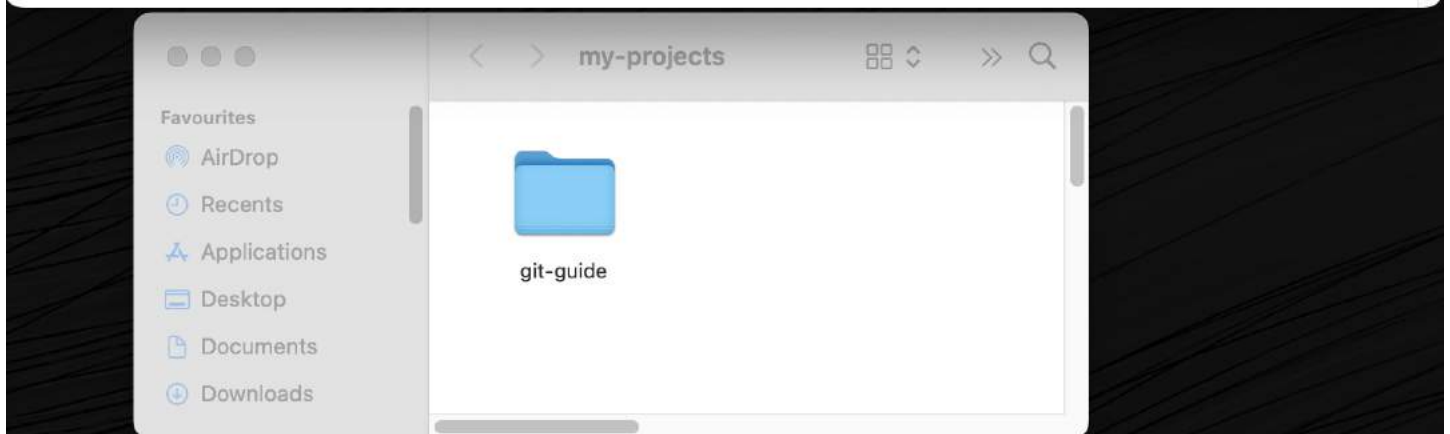- Amends and Rename
- Cherry Pick
- Force push

1. I want to clone a project and start going through the code.

- Get the repo URL/link
- Open your terminal
- Navigate to the location in the terminal where you want your project to be cloned
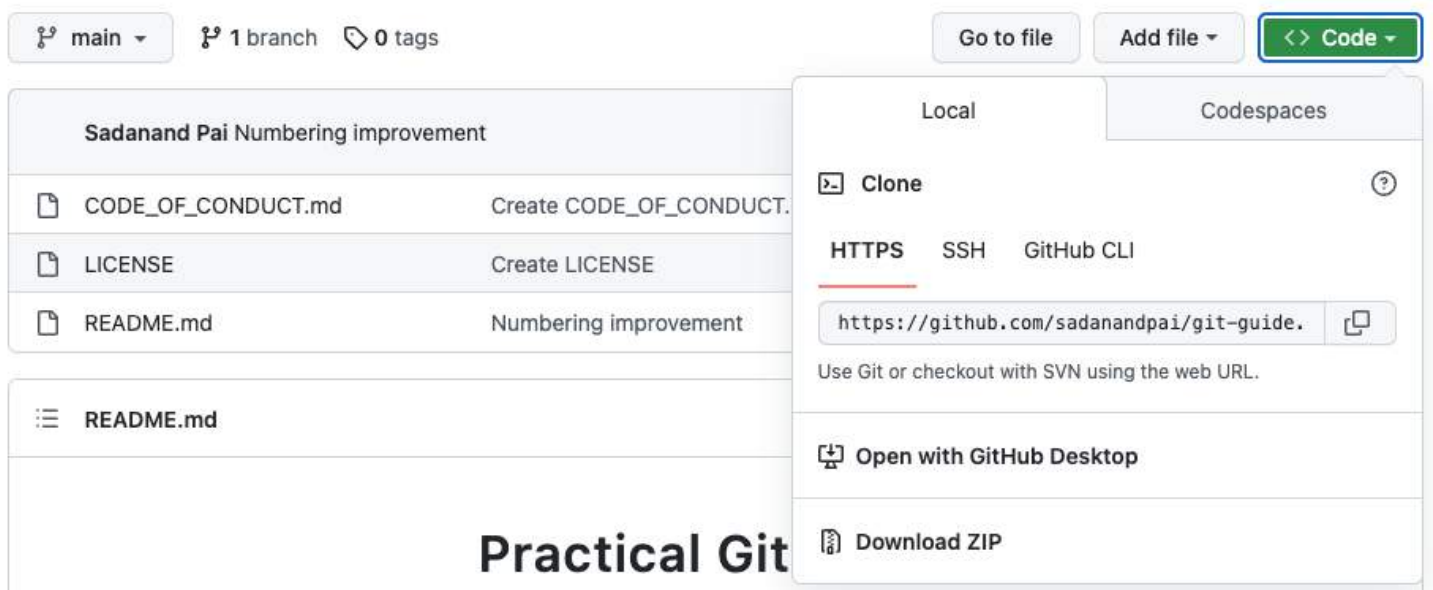- Enter the command `git clone URL`

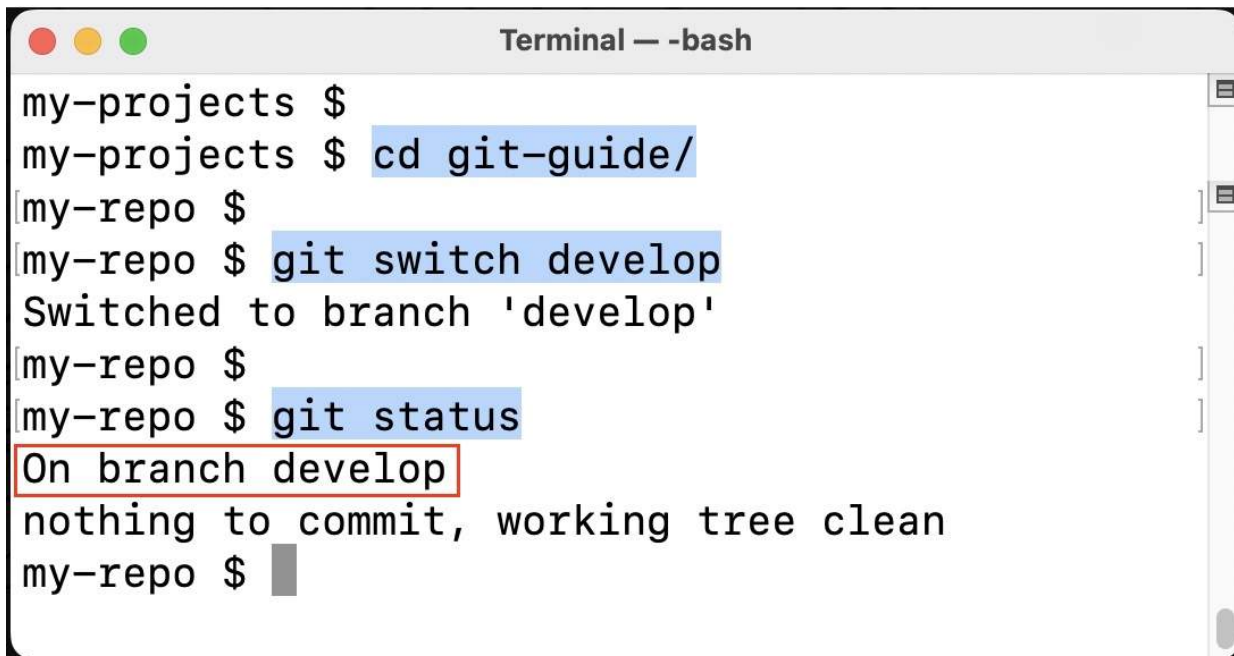The repo URL can be found inside the repo.

Github repo link example

2. I have cloned the repo. But I am not able to see the proper code!!!

By default the main/master branch is active. Ask which branch has the relevant code

- Navigate inside the cloned folder `cd repo<-name>`
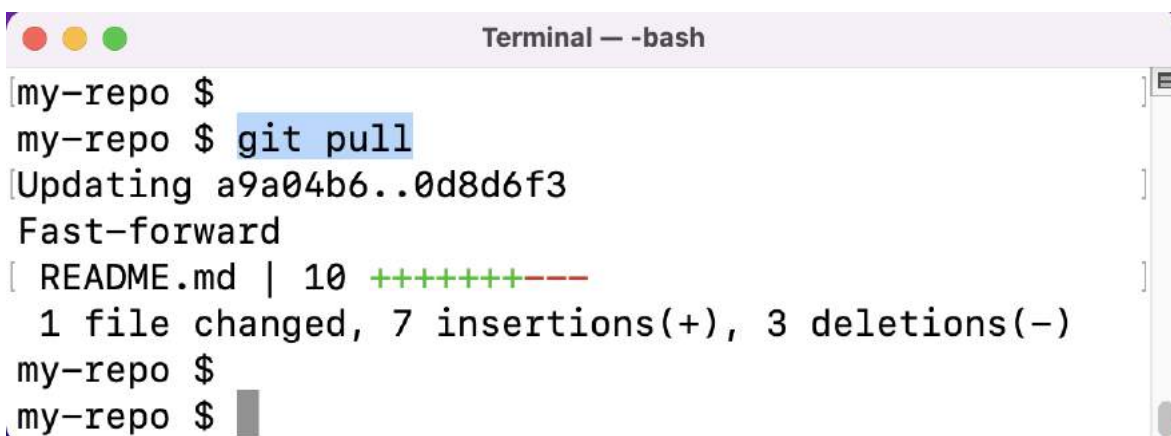- Enter the command `git switch <branch-name>`

```
 ● ● ●                  Terminal — -bash
my-projects $
my-projects $ cd git-guide/
[my-repo $
[my-repo $ git switch develop
 Switched to branch 'develop'
[my-repo $
[my-repo $ git status
On branch develop
nothing to commit, working tree clean
my-repo $ ▌
```

Note: The **checkout** command can also be used to switch the branch. `git checkout <branch-name>`

3. Someone has made a few changes in the code and asked me to pull those changes. What should I do?

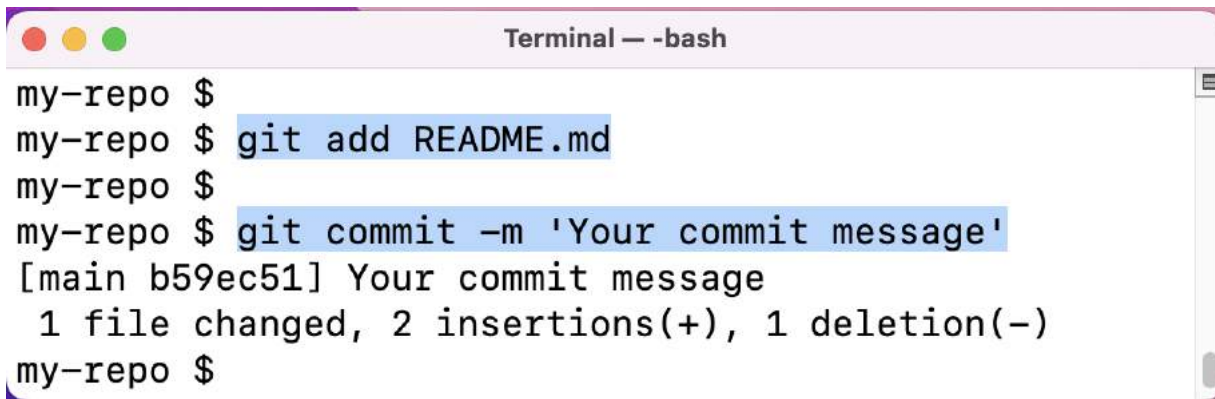- Pull the changes `git pull`

```
 ● ● ●                  Terminal — -bash
[my-repo $
my-repo $ git pull
[Updating a9a04b6..0d8d6f3
 Fast-forward
[ README.md | 10 +++++++---
  1 file changed, 7 insertions(+), 3 deletions(-)
my-repo $
my-repo $ ▌
```

4. I have modified the code and added changes. How do I commit my changes?

First stage all the files and then commit your changes. You can create multiple commits.

- Stage the files `git add *`
- Commit the changes `git commit -m 'Your commit message'`
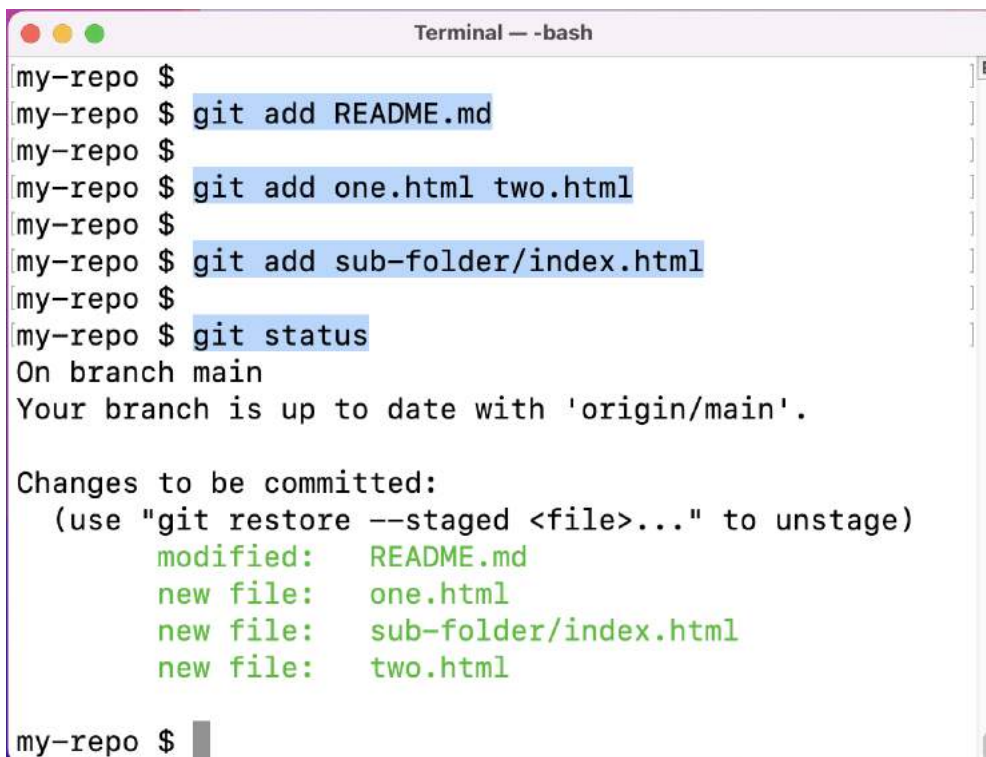
```
Terminal — -bash

my-repo $
my-repo $ git add README.md
my-repo $
my-repo $ git commit -m 'Your commit message'
[main b59ec51] Your commit message
 1 file changed, 2 insertions(+), 1 deletion(-)
my-repo $
```

5. What if I want only some of my files added & pushed instead of all changes?

You can add files by mentioning the file/files with relative or full path. You can add files one by one or multiple files at a time using the commands

- Add single file `git add <file-path>`
- Add multiple files `git add <file1-path> <file2-path>`
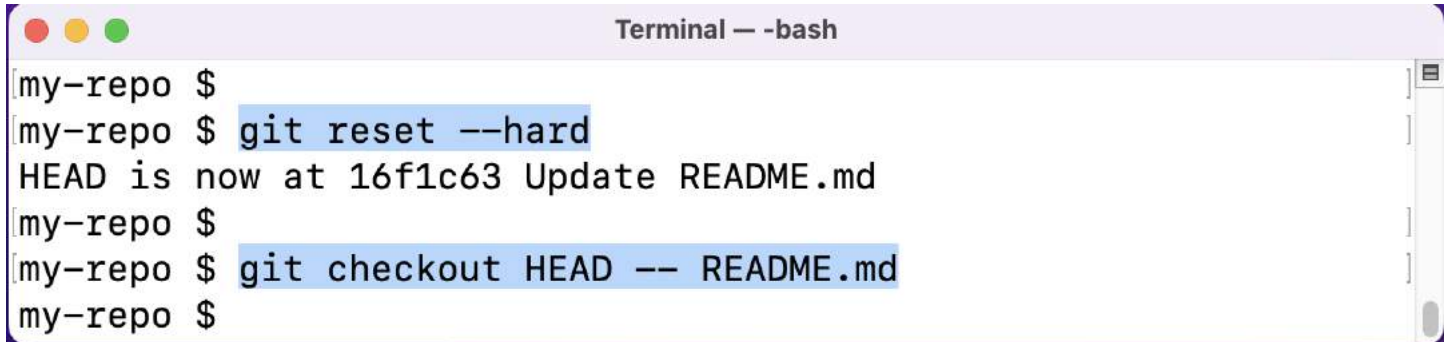
```
Terminal — -bash

my-repo $
my-repo $ git add README.md
my-repo $
my-repo $ git add one.html two.html
my-repo $
my-repo $ git add sub-folder/index.html
my-repo $
my-repo $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        new file:   one.html
        new file:   sub-folder/index.html
        new file:   two.html

my-repo $
```

Similarly, to unstage a file use the command `git reset <file>`

6. I have modified/formatted some code while going through it. Now I want the code to be back to the state as it was.

- To reset all the changes `git reset --hard`
- To reset a single file `git checkout HEAD -- <file-path>`

```
● ● ●                        Terminal — -bash
[my-repo $
[my-repo $ git reset --hard
HEAD is now at 16f1c63 Update README.md
[my-repo $
[my-repo $ git checkout HEAD -- README.md
my-repo $
```

7. I have committed my changes. How can I undo my change?

You can undo the commit by resetting the HEAD. If you just want to undo the commit but let the changes be present then use the `soft` attribute else if you do want the commit along with the changes then use the `hard` attribute

- `git reset --soft HEAD~1` (undo with changes preserved)
- `git reset --hard HEAD~1` (undo with changes removed)

8. I have made some changes to the branch. Also, I wanted to pull the new changes. But it is not working.

The command `git pull` may not work if the changes are done by someone else to the same files which you have also modified.

- Stash the changes `git stash save <name of the change> -u`
- Pull the changes now `git pull`
- Retrieve the changes `git stash apply <n>`

where n is the stash number. To get the list of stashes `git stash list`

```
●  ●  ●                      Terminal — -bash

my-repo $
my-repo $ git pull
Updating 3e6857c..66675d6
error: Your local changes to the following files would be overwritten by merge:
        README.md
Please commit your changes or stash them before you merge.
Aborting
my-repo $
my-repo $ git stash save 'My changes X' -u
Saved working directory and index state On main: My changes X
my-repo $
my-repo $ git pull
Updating 3e6857c..66675d6
Fast-forward
 README.md | 50 +++++++++++++++++++++++++++++---------------------------
 1 file changed, 25 insertions(+), 25 deletions(-)
my-repo $
my-repo $ git stash apply 0
Auto-merging README.md
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
my-repo $ █
```
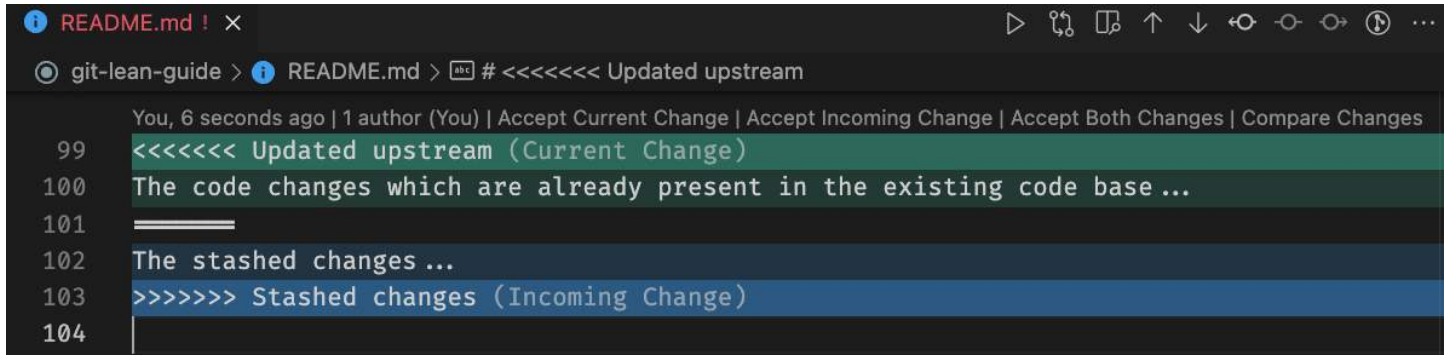
Note: You can stash multiple changes and bring them back as and when you like to

9. After applying the stash, I am getting a lot of conflicts in the code.

If there are changes in the code on the region of the stashed code, it is expected to get conflicts.
You will have to manually resolve all the conflicts. (Do it carefully)

```
●  ●  ●                  Terminal — -bash

my-repo $
my-repo $ git stash apply 0
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
my-repo $
```

Below is the view of the VSCode, which helps in easily resolving the conflicts.
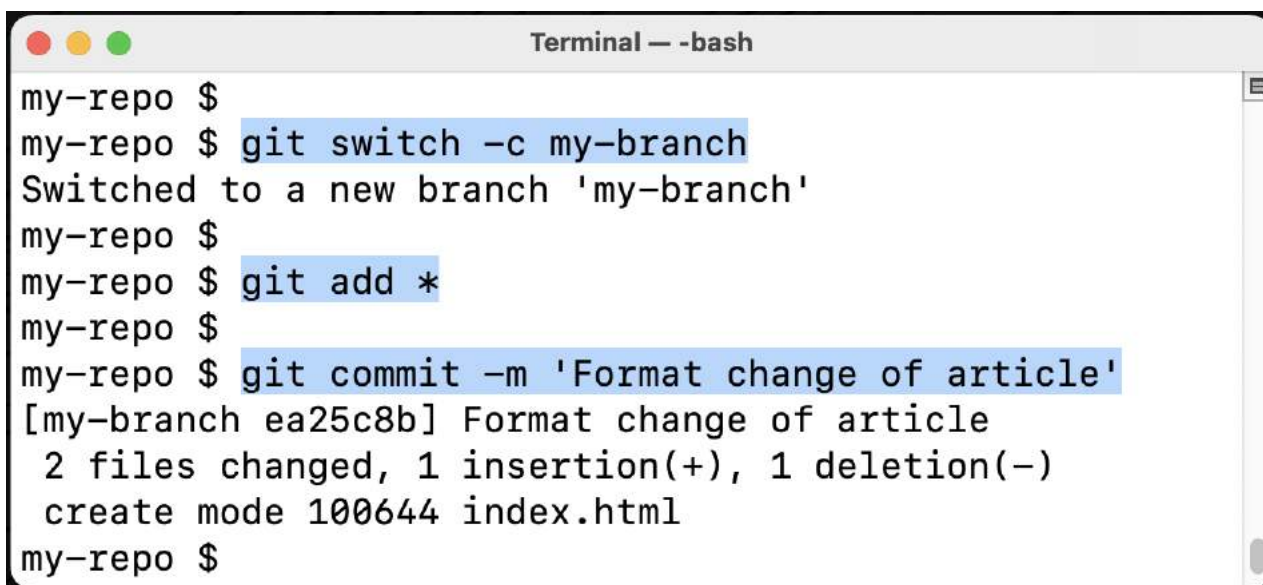
10. I have made some code changes. But I want to commit to a new separate branch.

You can create a separate branch out of the current branch and commit it. This works both if you have already made changes or are yet to start making changes.

- Create a new branch `git switch -c <my-branch-name>`
- Stage all the changes `git add *`
- Commit the changes `git commit -m "<some commit message>"`

To switch between the branches use the command `git checkout <original-branch-name>`

Alternatively, **checkout** command can also be used to create the branch. `git checkout -b <my-branch-name>`

Note: `my-branch-name` is your local branch and not available for anyone else unless you push it

11. I just committed but forgot to add a few files to the commit. Is there a way to update the same commit with some modifications?

Yes. You can update the commit by amending your changes.

- To add files `git add <file-name> <file-name> ...`
- To update the commit `git commit --amend --no-edit`

To update with new commit message `git commit --amend -m 'My new commit message'` (This command can also be used to update previous commit message without any code modifications)

`--no-edit` is used to avoid the prompt to edit commit message. If you wanna modify commit message as well, during commit update, then do not include `-m` along with your commit message

```
my-repo $
my-repo $ git add README.md
my-repo $
my-repo $ git commit --amend --no-edit
[main e635a2c] My old changes
 Date: Sat Jul 23 22:30:20 2022 +0530
 1 file changed, 7 insertions(+)
my-repo $
my-repo $ git commit --amend -m 'My new changes'
[main f0e915e] My new changes
 Date: Sat Jul 23 22:30:20 2022 +0530
 1 file changed, 7 insertions(+)
my-repo $ █
```

Note: The `amend` updates the previous commit without creating a new one on top of the previous. (in reality, Git discards the previous commit and replaces it with a new commit)

12. I am asked to raise a Pull Request (PR). What am I supposed to do?

You can follow the same steps as given in the previous question. Once done you will push the code and raise a PR. It's that simple. Here we assume you are on the 'develop' branch and raising PR to the 'main' branch.

- Create a new branch `git switch -c <my-branch-name>`
- Stage all the changes `git add *`
- Commit the changes `git commit -m "<some commit message>"`
- Push the changes `git push` (as the branch is not present on the remote, it will show the command to use)

- Enter `git push --set-upstream origin <my-branch-name>`

```
                        Terminal — -bash
my-repo $
my-repo $ git push
fatal: The current branch develop has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin develop

my-repo $
my-repo $ git push --set-upstream origin develop
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 280 bytes | 280.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/sadanandpai/git-lean-guide/pull/new/develop
remote:
To https://github.com/sadanandpai/git-lean-guide.git
 * [new branch]      develop -> develop
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
my-repo $
```

The URL to raise the PR will be automatically available as shown above. Use the link and open it in the browser.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: main ▾ ← compare: develop ▾ ✓ Able to merge. These branches can be automatically merged.

X Feature changes

Write | Preview   H B I ≔ <> 🔗 ≔ ≔ ☑ @ ↩

I have added features Y
I have modified feature Z

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Now select the base branch to which you want to raise a PR and click on 'Create a PR'
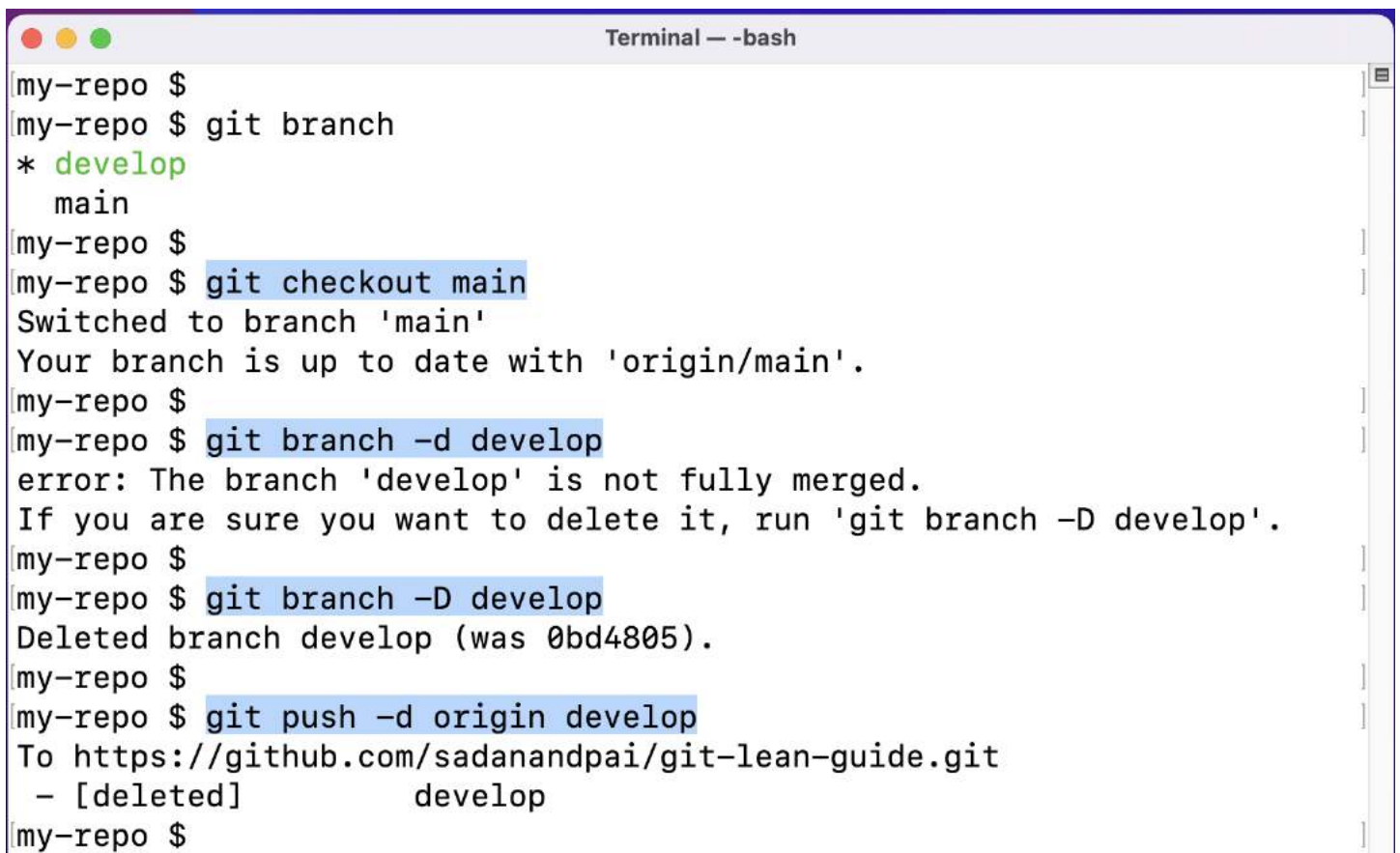
13. I do not want a branch anymore. How can I delete the branch?

To delete a branch locally, check out a different branch than the one you want to delete. Here we will delete the branch named 'develop'

- Switch to other branch `git checkout <other-branch>`
- Delete branch `git branch -d <branch-name>`

To delete the branch from remote as well

- Delete remote branch `git push -d <remote> <branch-name>`

```
[my-repo $
[my-repo $ git branch
* develop
  main
[my-repo $
[my-repo $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[my-repo $
[my-repo $ git branch -d develop
error: The branch 'develop' is not fully merged.
If you are sure you want to delete it, run 'git branch -D develop'.
[my-repo $
[my-repo $ git branch -D develop
Deleted branch develop (was 0bd4805).
[my-repo $
[my-repo $ git push -d origin develop
To https://github.com/sadanandpai/git-lean-guide.git
 - [deleted]         develop
[my-repo $
```
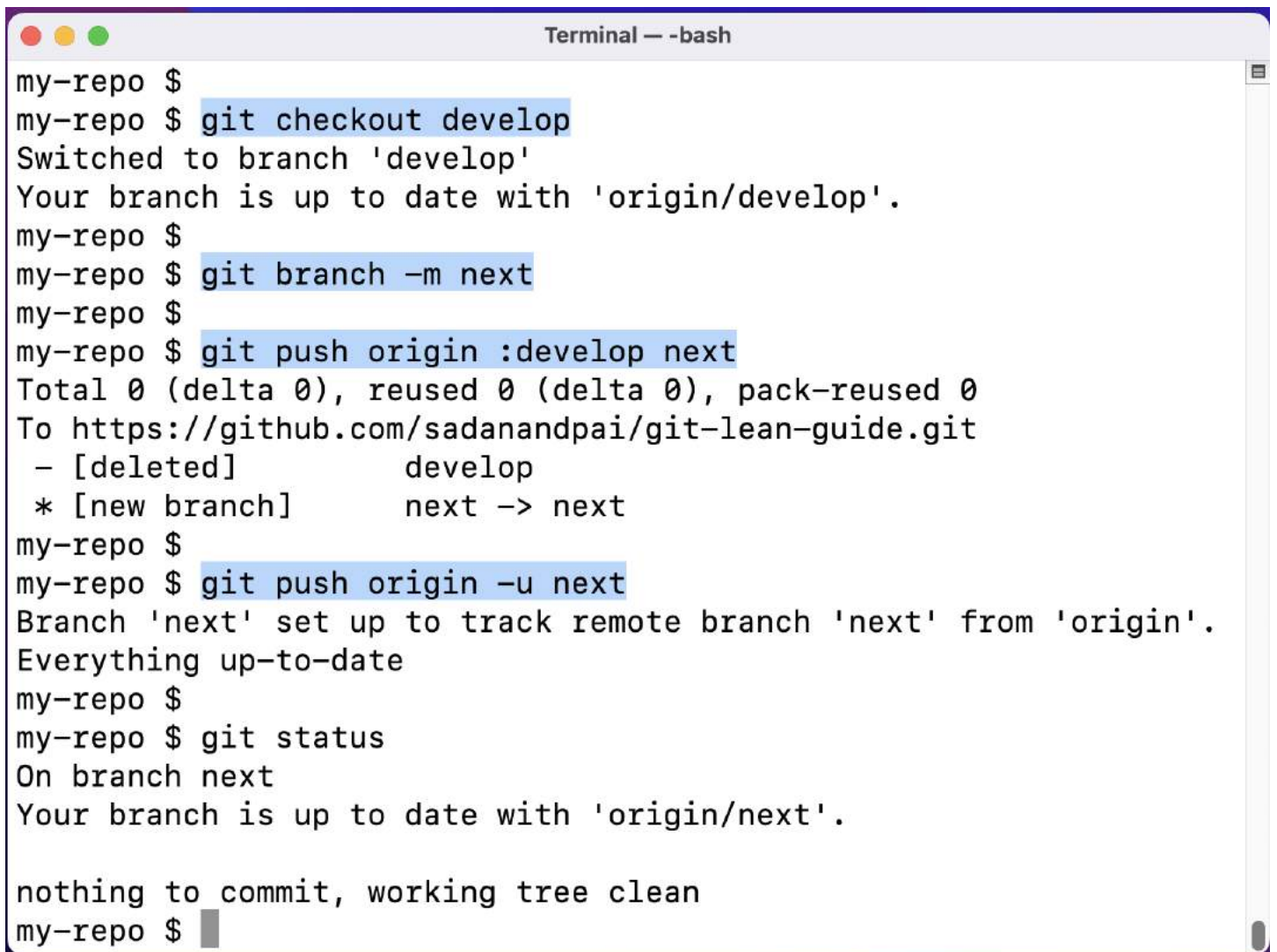
Note: If `-d` does not allow to delete a branch, use the `-D`. Example: `git branch -D <branch-name>`

14. I want to rename my local and remote branches. How can I do it?

To rename a branch, checkout to the branch and rename it.

- Check out other branch `git checkout <your-branch-name>`
- Rename your branch `git branch -m <new-branch-name>`
- Push to remote `git push <remote> :<your-branch> <new-branch-name>`
- Set upstream `git push <remote> -u <new-branch-name>`

<remote> is usually origin
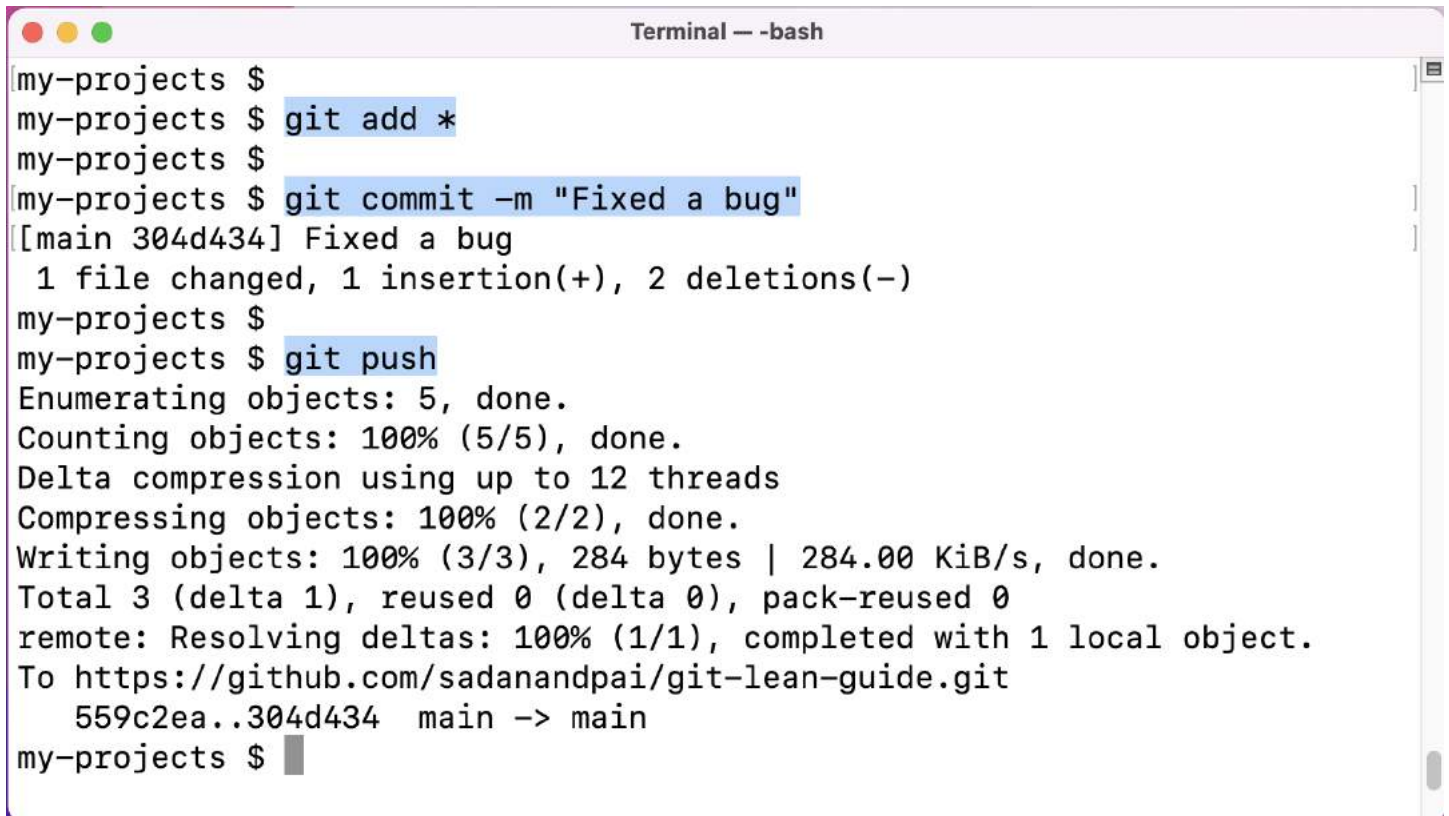
```
Terminal — -bash

my-repo $
my-repo $ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
my-repo $
my-repo $ git branch -m next
my-repo $
my-repo $ git push origin :develop next
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sadanandpai/git-lean-guide.git
 - [deleted]         develop
 * [new branch]      next -> next
my-repo $
my-repo $ git push origin -u next
Branch 'next' set up to track remote branch 'next' from 'origin'.
Everything up-to-date
my-repo $
my-repo $ git status
On branch next
Your branch is up to date with 'origin/next'.

nothing to commit, working tree clean
my-repo $ █
```

15. I have made some changes to the code on the branch on which all of the developers are working. How can I publish my changes?

To move the changes from your local machine to remote (called origin), follow the below steps in your terminal

- Stage the files `git add *`
- Commit the changes `git commit -m "<some commit message>"`
- Push the changes `git push`
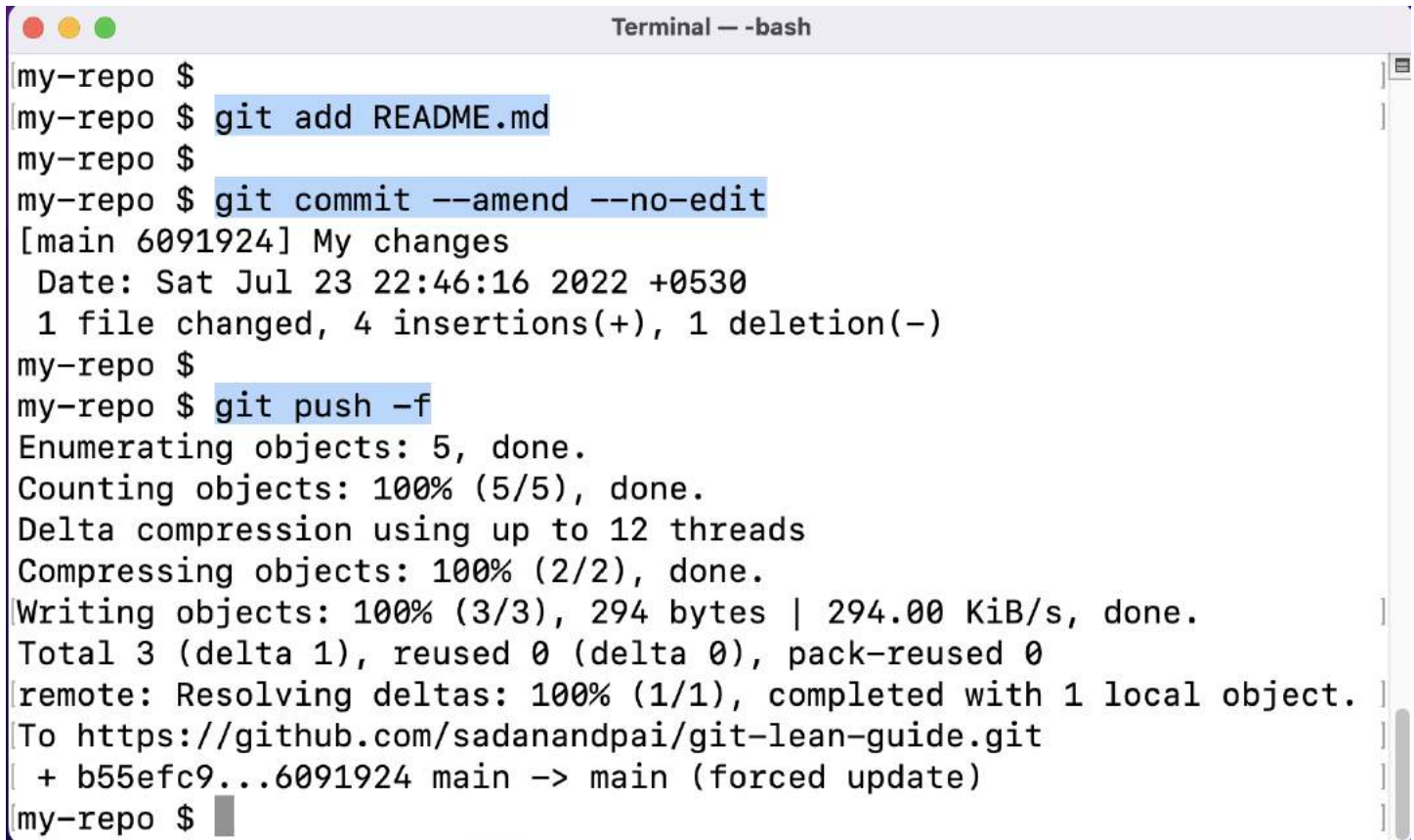
```
●●●                          Terminal — -bash
[my-projects $
my-projects $ git add *
my-projects $
[my-projects $ git commit -m "Fixed a bug"
[[main 304d434] Fixed a bug
 1 file changed, 1 insertion(+), 2 deletions(-)
my-projects $
my-projects $ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 284 bytes | 284.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sadanandpai/git-lean-guide.git
   559c2ea..304d434  main -> main
my-projects $ ▊
```

16. I created a commit and also pushed it. Is it possible to update that commit now?

Yes. You can update the commit even after it is pushed. Everything will follow as mentioned in the previous question, but you will have to force push.

- `git add <file-name> <file-name> ...`
- `git commit --amend --no-edit` or `git commit --amend`
- `git push -f` or `git push --force`

```
Terminal — -bash

my-repo $
my-repo $ git add README.md
my-repo $
my-repo $ git commit --amend --no-edit
[main 6091924] My changes
 Date: Sat Jul 23 22:46:16 2022 +0530
 1 file changed, 4 insertions(+), 1 deletion(-)
my-repo $
my-repo $ git push -f
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 294.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sadanandpai/git-lean-guide.git
 + b55efc9...6091924 main -> main (forced update)
my-repo $
```

Note: You need to be very careful while pushing forcefully, as it may eliminate other commits if someone has done in between. Make sure you are working on the branch and no one else is simultaneously working on the same or branching out from the branch at your commit.

17. I have created single/multiple commits. When I am trying to push my changes, getting a rejected message. I am stuck!!!

The rejection could be because the remote branch might be ahead of the local branch. Different techniques can be used here to achieve sync.

- Pull the changes `git pull`
- Continue with the merge commit created automatically
- `git push`

```
●●●                      Terminal — -bash

my-repo $
my-repo $ git add README.md
my-repo $
my-repo $ git commit -m 'Updates in Readme'
[main 225c355] Updates in Readme
 1 file changed, 2 insertions(+)
my-repo $
my-repo $ git push
To https://github.com/sadanandpai/git-lean-guide.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/sadanandpai/git-lean-guide.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
my-repo $
my-repo $ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 647 bytes | 161.00 KiB/s, done.
From https://github.com/sadanandpai/git-lean-guide
   e39c9e2..311172f  main        -> origin/main
Auto-merging README.md
Merge made by the 'recursive' strategy.
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
my-repo $
my-repo $ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 605 bytes | 302.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/sadanandpai/git-lean-guide.git
   311172f..31fb0fd  main -> main
my-repo $ ▎
```

If there are conflicts, then resolve them manually to proceed ahead as shown below.

18. I followed the above steps but got conflicts after git pull.

If there are code changes on the same region from multiple commits, conflicts will occur. You need to resolve all the conflicts and proceed.

- Resolve all the conflicts
- Stage files `git add <file-path>`
- Continue the merge `git merge --continue`
- Push the changes `git push`

```
                          Terminal — -bash

my-repo $
my-repo $ git push
To https://github.com/sadanandpai/git-lean-guide.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/sadanandpai/git-lean-guide.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
my-repo $ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 645 bytes | 161.00 KiB/s, done.
From https://github.com/sadanandpai/git-lean-guide
   434ec72..a6d46c3  main          -> origin/main
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
my-repo $
my-repo $ git add README.md
my-repo $
my-repo $ git merge --continue
[main 6bf88dd] Merge branch 'main' of https://github.com/sadanandpai/git-lean-guide
my-repo $
my-repo $ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 584 bytes | 292.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/sadanandpai/git-lean-guide.git
   a6d46c3..6bf88dd  main -> main
my-repo $
```
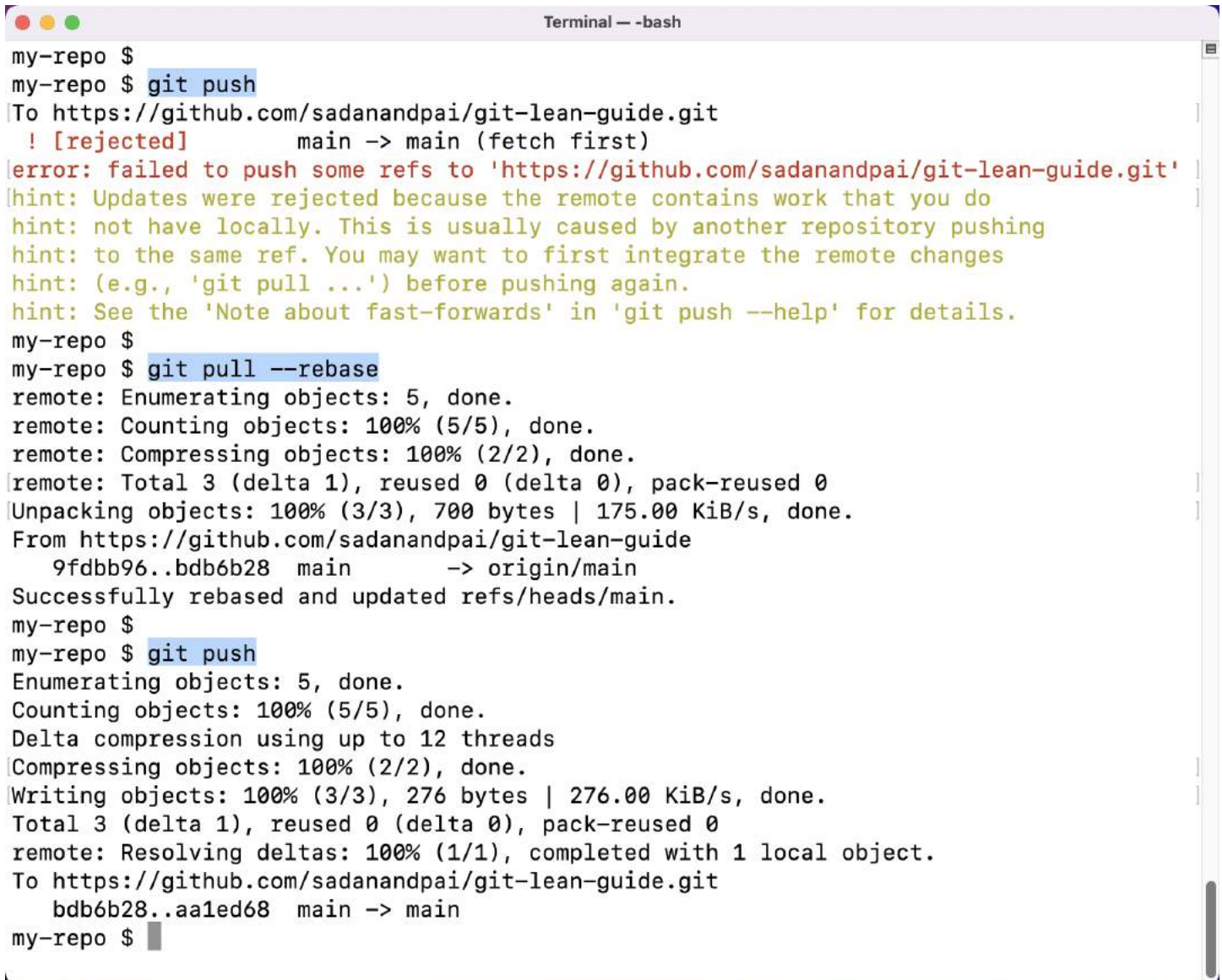
Note: If something goes wrong, in any of the above steps, then there is nothing to panic about. Just run `git merge --abort` and redo the steps.

19. I have created single/multiple commits. When I am trying to push my changes, getting a rejected message. Can I pull the new changes without merging the commit (Rebase)?

Yes. You can pull the changes without a merge. This is called **Rebase**. I know you have heard it a lot. It is very simple though.

- Pull with rebase `git pull --rebase`
- Push the changes `git push`

```
my-repo $
my-repo $ git push
To https://github.com/sadanandpai/git-lean-guide.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/sadanandpai/git-lean-guide.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
my-repo $
my-repo $ git pull --rebase
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 700 bytes | 175.00 KiB/s, done.
From https://github.com/sadanandpai/git-lean-guide
   9fdbb96..bdb6b28  main         -> origin/main
Successfully rebased and updated refs/heads/main.
my-repo $
my-repo $ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 276 bytes | 276.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sadanandpai/git-lean-guide.git
   bdb6b28..aa1ed68  main -> main
my-repo $
```

If you get conflicts, then solve all the conflicts. Then

- `git rebase --continue`
- Resolve all conflicts
- Push the changes `git push`

```
●●●                          Terminal — -bash

my-repo $
my-repo $ git push
To https://github.com/sadanandpai/git-lean-guide.git
 ! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'https://github.com/sadanandpai/git-lean-guide
.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
my-repo $
my-repo $ git pull --rebase
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
error: could not apply 99c7cbb... x
Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted_files>", then run "git rebase --continue".
You can instead skip this commit: run "git rebase --skip".
To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 99c7cbb... x
my-repo $
my-repo $ git rebase --continue
[detached HEAD 9fdbb96] Rebase flow
 1 file changed, 19 insertions(+), 6 deletions(-)
Successfully rebased and updated refs/heads/main.
my-repo $
my-repo $ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 486 bytes | 486.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sadanandpai/git-lean-guide.git
   2a4deea..9fdbb96  main -> main
my-repo $ ▊
```

Note: If something goes wrong, in any of the above steps, then there is nothing to panic about. Just run `git rebase --abort` and redo the steps.

20. I have raised a PR (Pull Request). But it is showing conflicts.

Minor improvements #1

⥮ Open sadanandpai wants to merge 1 commit into `main` from `develop` ⧉

💬 Conversation 0	⊶ Commits 1	▣ Checks 0	± Files changed 1

> **sadanandpai** commented 34 seconds ago   ☺ ⋯
>
> *No description provided.*

⊶   Minor improvements   c2db429

Add more commits by pushing to the **develop** branch on **sadanandpai/git-lean-guide**.

⚠ **This branch has conflicts that must be resolved**   **Resolve conflicts**
Use the web editor or the command line to resolve conflicts.
Conflicting files

README.md

Merge pull request ▾	You can also open this in GitHub Desktop or view command line instructions.

PR will show conflicts if the new changes added to the source branch are conflicting with your changes or your branch is lagging.

There are 2 main approaches to solve this.

- Merge approach
- Rebase approach

Follow any one of the approaches. Don't try both of them.

Assuming that your branch is `develop` and the source branch is `main`

Merge approach

- Checkout to main branch `git checkout main`
- Pull changes `git pull`

- Checkout to your branch `git checkout develop`
- Merge the changes `git merge main`
- Resolve all the conflicts and add to staging `git add <files>`
- If conflicts are present `git merge --continue`
- Push the changes `git push`

```
Terminal — -bash
[my-repo $
[my-repo $ git checkout main
Switched to branch 'main'
Your branch is behind 'origin/main' by 20 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
[my-repo $
[my-repo $ git pull
Updating c6c284e..0450e18
Fast-forward
  README.md  | 259 +++++++++++++++++++++++++++++++++++++++++++++---------------
  index.html |   1 -
  2 files changed, 183 insertions(+), 77 deletions(-)
  delete mode 100644 index.html
[my-repo $
[my-repo $ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
[my-repo $
[my-repo $ git merge main
Removing index.html
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
[my-repo $
[my-repo $ git add README.md
[my-repo $
my-repo $ git merge --continue
[develop 32a2e45] Merge branch 'main' into develop
[my-repo $
[my-repo $ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 379 bytes | 189.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sadanandpai/git-lean-guide.git
   c2db429..32a2e45  develop -> develop
my-repo $ ▮
```

Rebase approach

- Checkout to main branch `git checkout main`
- Pull changes `git pull`

- Checkout to your branch `git checkout develop`
- Rebase the branch `git rebase main`
- Resolve all the conflicts and add to staging `git add <files>`
- Run `git rebase --continue` after resolving the conflicts
- Push the changes `git push -f`

```
●●●                    Terminal — -bash
my-repo $
my-repo $ git checkout main
Switched to branch 'main'
Your branch is behind 'origin/main' by 5 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
my-repo $
my-repo $ git pull

Updating 2a4deea..0450e18
Fast-forward
 README.md | 97 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++-------------
 1 file changed, 79 insertions(+), 18 deletions(-)
my-repo $
my-repo $ git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
my-repo $
my-repo $ git rebase main
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
error: could not apply 7a836f7... x
Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted_files>", then run "git rebase --continue".
You can instead skip this commit: run "git rebase --skip".
To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 7a836f7... x
my-repo $
my-repo $ git add README.md
my-repo $
my-repo $ git rebase --continue
[detached HEAD 3cbf6ae] Minor improvements
 1 file changed, 4 deletions(-)
Successfully rebased and updated refs/heads/develop.
my-repo $
my-repo $ git push -f
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 295 bytes | 147.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sadanandpai/git-lean-guide.git
 + 7a836f7...3cbf6ae develop -> develop (forced update)
my-repo $
```

Note: You may have to run `git rebase --continue` multiple times if there are multiple conflicts on your multiple commits.

21. I have many commits. How can I transform them into a single commit (squash)?

Converting multiple commits into one is known as **Squashing**. We will achieve this by rebasing with the help of the interactive feature of the editor (VSCode). This will be both easier and simple.

- Run `git rebase -i HEAD~<n>` (where n refers to the number of commits to squash)
- Mark all the commits as 'Squash' except the oldest one
- Click on 'Start rebase'
- Enter the commit message

Note: If you had already pushed the commits, then you will have to use the command `git push -f` to reflect squash on the remote branch as well.

22. I have many commits. How can I transform them into a single commit (squash) with just commands?

You can use the technique of undoing to achieve this easily.

- Undo the number of commits you need to squash `git reset --soft HEAD~<n>`
- Commit them again `git commit -m 'Commit message'`

Note: If you had already pushed the commits, then you will have to use the command `git push -f` to reflect squash on the remote branch as well.
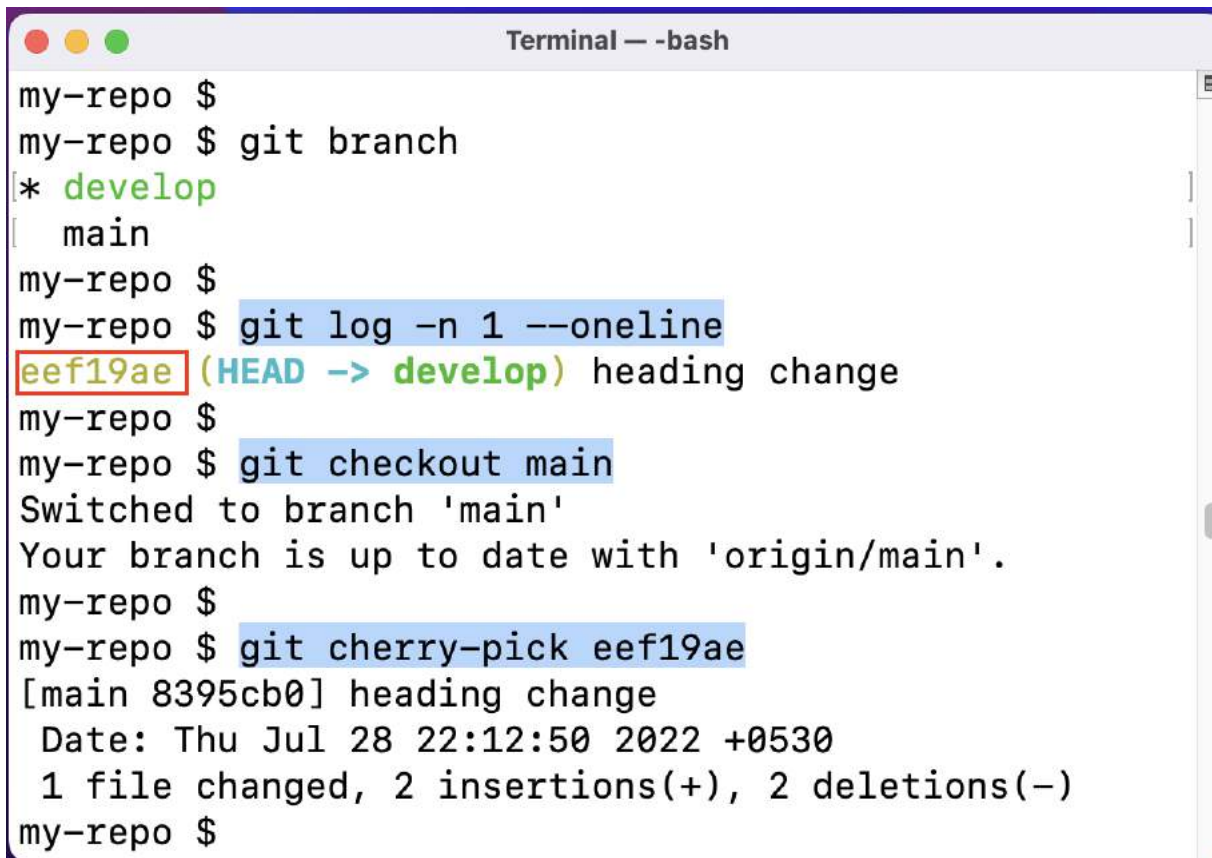
23. I am trying to rebase my branch with the same or another branch. As I have many commits, I am getting a lot of conflicts on every commit to rebase.

- You can first squash all commits to one commit (Refer 21).
- Once done, rebase the branch and resolve all conflicts in a single go. (Refer 20)

24. I have made changes and committed to a branch. I want to copy the same changes to another branch.

To copy the changes of a commit from one branch to another, you can use **cherry pick**. First obtain the commit id of the commit, which you want to copy.

- Checkout the branch `git checkout <destination-branch>`
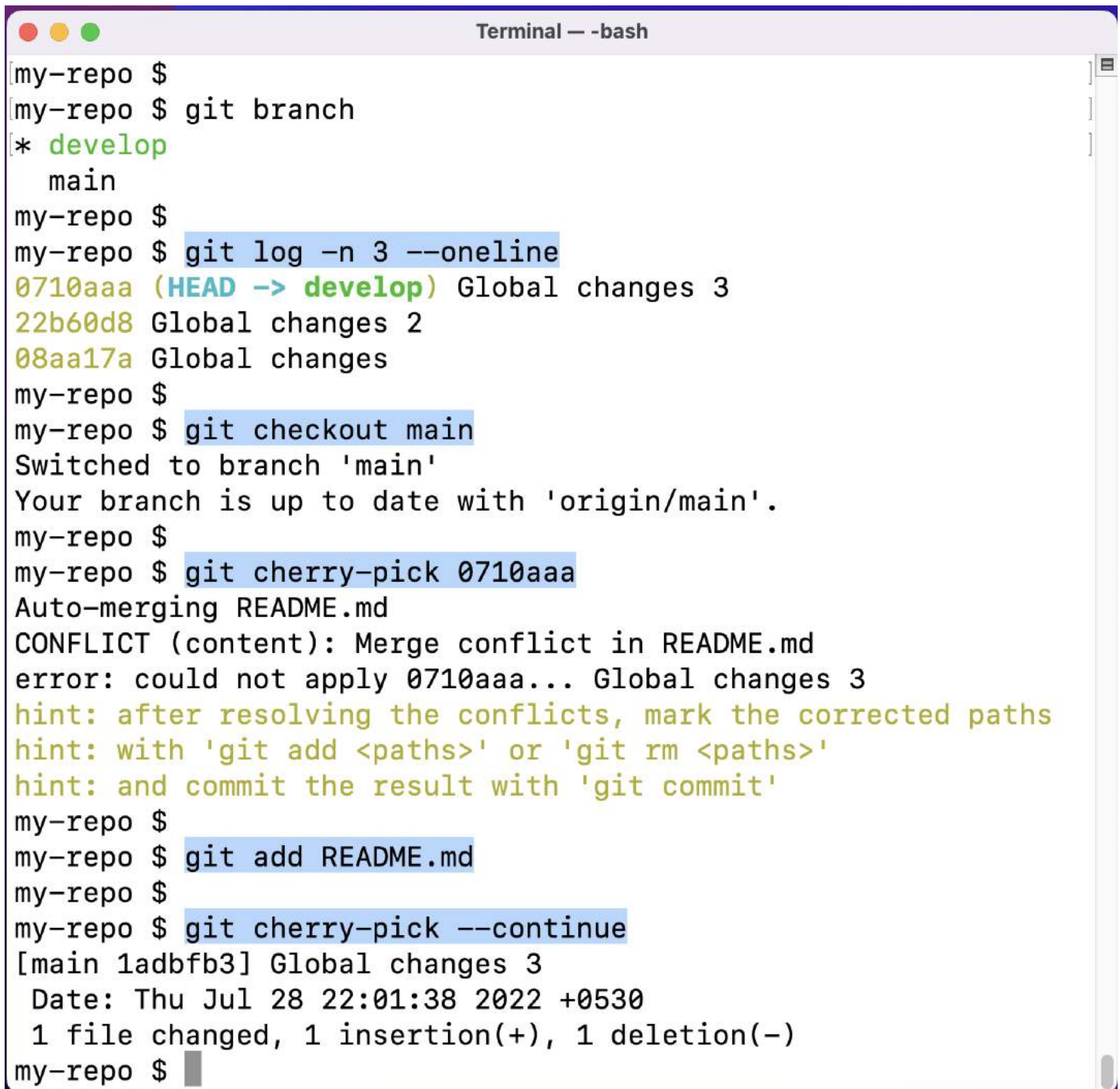- Cherry-pick the commit `git cherry-pick <commit-id>`

```
Terminal — -bash

my-repo $
my-repo $ git branch
* develop
  main
my-repo $
my-repo $ git log -n 1 --oneline
eef19ae (HEAD -> develop) heading change
my-repo $
my-repo $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
my-repo $
my-repo $ git cherry-pick eef19ae
[main 8395cb0] heading change
 Date: Thu Jul 28 22:12:50 2022 +0530
 1 file changed, 2 insertions(+), 2 deletions(-)
my-repo $
```

25. I have tried to cherry-pick as shown in 24. But I am getting conflicts.

If you get any conflicts, resolve the conflicts first. Once all the conflicts are resolved, they continue the cherry-pick.

- Add resolved files to stage `git add <file-paths>`
- Continue cherry-pick `git cherry-pick --continue`

```
[my-repo $
[my-repo $ git branch
[* develop
   main
my-repo $
my-repo $ git log -n 3 --oneline
0710aaa (HEAD -> develop) Global changes 3
22b60d8 Global changes 2
08aa17a Global changes
my-repo $
my-repo $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
my-repo $
my-repo $ git cherry-pick 0710aaa
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
error: could not apply 0710aaa... Global changes 3
hint: after resolving the conflicts, mark the corrected paths
hint: with 'git add <paths>' or 'git rm <paths>'
hint: and commit the result with 'git commit'
my-repo $
my-repo $ git add README.md
my-repo $
my-repo $ git cherry-pick --continue
[main 1adbfb3] Global changes 3
 Date: Thu Jul 28 22:01:38 2022 +0530
 1 file changed, 1 insertion(+), 1 deletion(-)
my-repo $
```

Note: If something goes wrong in between, reset the process by using the command `git cherry-pick --abort` (You can start the process again)

26. I have multiple commits which I want to move to a different branch.

To copy a range of commits from one branch to another, you can note down the older commit id from history and the newer commit id.
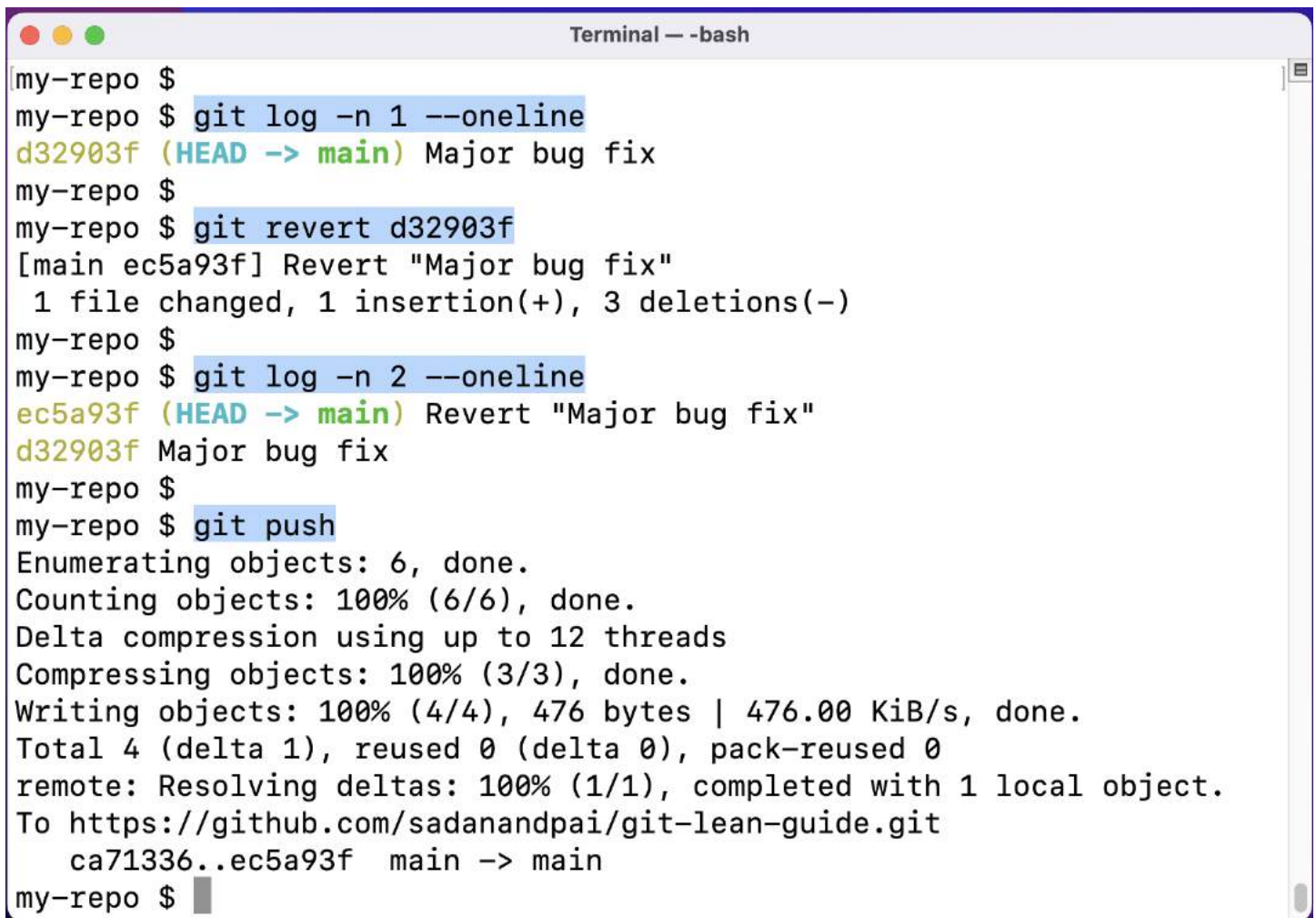
- Checkout the branch `git checkout <destination-branch>`
- Cherry pick the commit `git cherry-pick <old-commit-id>..<new-commit-id>`

```
[my-repo $
[my-repo $ git log -n 3 --oneline
0710aaa (HEAD -> develop) Global changes 3
22b60d8 Global changes 2
08aa17a Global changes
[my-repo $
[my-repo $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[my-repo $
[my-repo $ git cherry-pick 08aa17a..0710aaa
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
error: could not apply 22b60d8... Global changes 2
hint: after resolving the conflicts, mark the corrected paths
hint: with 'git add <paths>' or 'git rm <paths>'
hint: and commit the result with 'git commit'
[my-repo $
[my-repo $ git add README.md
[my-repo $
[my-repo $ git cherry-pick --continue
[main a1cad1a] Global changes 2
 Date: Thu Jul 28 22:01:27 2022 +0530
 1 file changed, 4 insertions(+), 1 deletion(-)
[main 4069ced] Global changes 3
 Date: Thu Jul 28 22:01:38 2022 +0530
 1 file changed, 1 insertion(+)
my-repo $ 
```

27. I have pushed my changes and got it merged. I want to revert it immediately.

Revert creates a reverse commit where the changes made will reverse and is created as a new commit. To revert a particular commit first, obtain its commit id.

- Revert the commit `git revert <commit-id>`
- Push the commit `git push`

```
[my-repo $
my-repo $ git log -n 1 --oneline
d32903f (HEAD -> main) Major bug fix
my-repo $
my-repo $ git revert d32903f
[main ec5a93f] Revert "Major bug fix"
 1 file changed, 1 insertion(+), 3 deletions(-)
my-repo $
my-repo $ git log -n 2 --oneline
ec5a93f (HEAD -> main) Revert "Major bug fix"
d32903f Major bug fix
my-repo $
my-repo $ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 476 bytes | 476.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sadanandpai/git-lean-guide.git
   ca71336..ec5a93f  main -> main
my-repo $ ▊
```

28. How do I reset the code of my branch to the code of a different branch?

When resetting a branch to a different branch, the codebase on your branch will become the same as the other branch. To achieve this, you can use the 'reset' command.

- Checkout to the branch `git checkout <your-branch-bame>`
- Reset with the branch name `git reset --hard <source-branch>`

In this case, 'your-branch-name' will match the codebase of 'source-branch'

```
Terminal — -bash

my-repo $
my-repo $ git checkout develop
Switched to branch 'develop'
my-repo $
my-repo $ git reset --hard main
HEAD is now at ec5a93f Revert "Major bug fix"
my-repo $
```

29. I want to delete/undo the previous commit from my branch which I have already pushed. I am not looking for a revert. I just want to delete it.

Revert will reverse the changes creating a new commit. If you want to remove the previous commit, then you can undo and force push the branch.

- Reset by undoing `git reset --hard HEAD~<n>` (n is the number of commits to undo)
- Push the changes `git push -f`

```
Terminal — -bash

my-repo $
my-repo $ git log -n 2 --oneline
665a35f (HEAD -> main, origin/main, origin/HEAD) Fix 2
d80b71c Fix 1
my-repo $
my-repo $ git reset --hard HEAD~1
HEAD is now at d80b71c Fix 1
my-repo $
my-repo $ git push -f
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sadanandpai/git-lean-guide.git
 + 665a35f...d80b71c main -> main (forced update)
my-repo $
my-repo $ git log -n 1 --oneline
d80b71c (HEAD -> main, origin/main, origin/HEAD) Fix 1
my-repo $
```

Note: Force push is needed as the history of the branch is changing in this. If the commit is not pushed then it is not needed.

Created by Sadanand Pai
Repo link: Practical Git Guide