

# Git Branches

What is the use of **Branches**?

- When adding a feature or repairing a bug then always create a new branch.
- Never commit directly to the “main” or “master” branch which is the default branch of a GitHub Repository. Because in Open-Source Projects, code in “main” is used by everyone, and maybe our code may have some errors and is not finalised yet.

## Head Branch

- It is the active and the checked out branch, commits are always made in Head Branch
- By default “HEAD” points to the “main” or “master” branch. Generally, commits are made on the “HEAD” inside the “.git” folder.

## Local Branches

- The branches present in the local machine.

## Remote Branches

- The branches present in remote location.

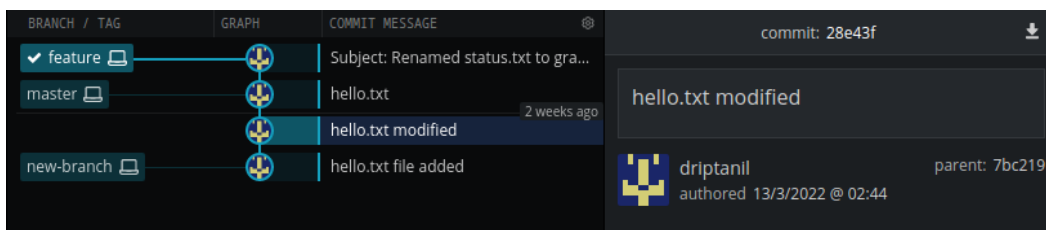
## Creating new Branch

- `git branch <branch_name>` creates a new local branch.

```
driptanil@driptanil in repo: learningGit on 0 master took 3ms
λ git branch feature
```

- `git branch <branch_name> <revision_id>` creates a new local branch at the commit with revision\_id

```
driptanil@driptanil in repo: LearningGit on 0 feature took 1ms
λ git branch new-branch 7bc219
```



## Switching Branches

- `git checkout <branch>` or `git switch <branch>` is used to change “HEAD” which will now point to the branch.

```
driptanil@driptanil in repo: learningGit on 0 master took 11ms
λ git checkout feature
Switched to branch 'feature'

driptanil@driptanil in repo: learningGit on 0 feature took 3ms
```

- Now commits are made to the checked out branch and the “main” or “master” branch remains unchanged.

"images/git21.png" is not created yet. Click to create.

"images/git22.png" is not created yet. Click to create.

## Renaming Branches

### Local Branches

- `git branch -m <new_branch>` will rename the current head branch.

```
driptanil@driptanil in repo: LearningGit on  feature took 13ms
λ git branch -m new-feature

driptanil@driptanil in repo: LearningGit on  new-feature took 13ms
λ git branch
master
new-branch
* new-feature
```

- `git branch -m <branch> <new_branch>` will rename the branch.

```
driptanil@driptanil in repo: LearningGit on  new-feature took 32ms
λ git branch -m new-branch old-branch

driptanil@driptanil in repo: LearningGit on  new-feature took 31ms
λ git branch
master
* new-feature
old-branch
```

### Remote Branches

## Git Merge

- `git merge <branch>` is used to update and merge all the changes made in the “feature” branch to the “main” or “master” branch.

"images/git23.png" is not created yet. Click to create.

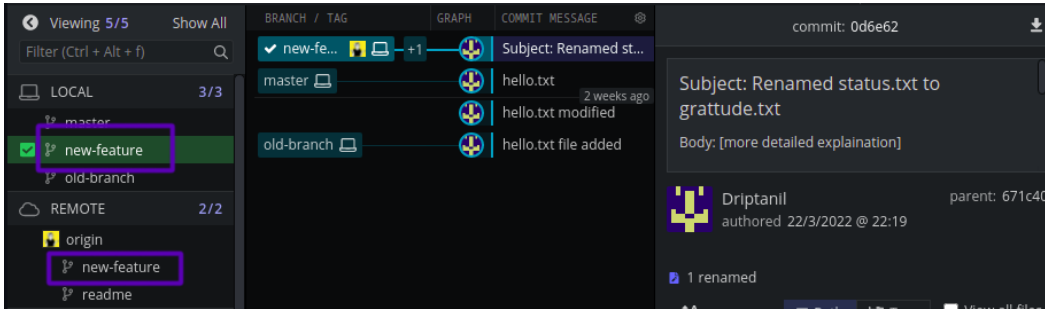
## Publishing Branches

- `git push -u origin <branch>` will push the local branch to remote.  
(`-u` helps establish a tracking connection)

```
driptanil@driptanil in repo: LearningGit on  master [?] took 7ms
λ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master

driptanil@driptanil in repo: LearningGit on  master [?] took 2ms
[●] * git push -u origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:driptanil/LearningGit
28e43fd..671c408 master → master
branch 'master' set up to track 'origin/master'.
```



## Tracking Connection

- By default, branches in git have no relation with each other.
- `git checkout --track <remote_branch>` helps establish a tracking connection and adds the remote branch to locally.

```
driptanil@driptanil in ~
λ cd Documents/LearningGit

driptanil@driptanil in repo: LearningGit on ♀ new-feature
λ git branch
master
* new-feature
  old-branch

driptanil@driptanil in repo: LearningGit on ♀ new-feature took 9ms
λ git checkout --track readme
fatal: missing branch name; try -b

driptanil@driptanil in repo: LearningGit on ♀ new-feature took 3ms
[●] * git checkout --track origin/readme
branch 'readme' set up to track 'origin/readme'.
Switched to a new branch 'readme'

driptanil@driptanil in repo: LearningGit on ♀ readme took 3ms
λ git branch
master
new-feature
old-branch
* readme
```

- `git branch --track <branch_name> <remote_branch>` helps establish a tracking connection which allows branch to track remote branch.

```
driptanil@driptanil in repo: LearningGit on ♀ readme took 2ms
[●] * git branch --track not-feature origin/new-feature
branch 'not-feature' set up to track 'origin/new-feature'.

driptanil@driptanil in repo: LearningGit on ♀ readme took 5ms
λ git branch
master
new-feature
not-feature
old-branch
* readme
```

- It allows `git push` & `git pull` in place of `git push origin <branch>` and also informs about unpulled and unpushed files using `git branch -v`.

## Pulling Branch

- `git pull` will pull all the changes in remote branch to local branch.

## Pushing Branch

- `git push` will push all the changes in local branch to remote branch.

```
driptanil@driptanil in repo: LearningGit on  master [↑1] took 13s
λ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 327 bytes | 327.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:driptanil/LearningGit
671c408..28bfa63 master -> master
```

## Deleting Branch

- `git branch -d <branch>` will delete the branch, but it will not remove the branch which is currently head.

```
driptanil@driptanil in repo: LearningGit on  not-feature took 4ms
[●] × git branch -d old-branch
Deleted branch old-branch (was 7bc219b).
```

```
driptanil@driptanil in repo: LearningGit on  not-feature took 11ms
λ git branch -d not-feature
error: Cannot delete branch 'not-feature' checked out at '/home/driptanil/Documents/LearningGit'
```

- `git branch -D <branch>` will force delete the branch with not pushed files.

```
driptanil@driptanil in repo: LearningGit on  new-feature [+] took 3ms
λ git status
On branch new-feature
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   nothing.txt

driptanil@driptanil in repo: LearningGit on  new-feature [+] took 2ms
λ git commit -m "nothing.txt created"
[new-feature 6e99610] nothing.txt created
1 file changed, 1 insertion(+)
create mode 100644 nothing.txt

driptanil@driptanil in repo: LearningGit on  new-feature took 7ms
λ git switch not-feature
Switched to branch 'not-feature'
Your branch is up to date with 'origin/new-feature'.

driptanil@driptanil in repo: LearningGit on  not-feature took 3ms
λ git branch -d new-feature
error: The branch 'new-feature' is not fully merged.
If you are sure you want to delete it, run 'git branch -D new-feature'.

driptanil@driptanil in repo: LearningGit on  not-feature took 2ms
[●] × git branch -D new-feature
Deleted branch new-feature (was 6e99610).
```

- `git push origin --delete <branch>` will remove delete remote branches.

Viewing 6/6		Show All	BRANCH / TAG	GRAPH	COMMIT MESSAGE	AUTHOR
Filter (Ctrl + Alt + f)			master		name.txt created	Driptanil 6 days ago
LOCAL 3/3			✓ not-feature	+2	Subject: Renamed status.txt to grat...	Driptanil
master					hello.txt	Driptanil 2 weeks ago
not-feature					hello.txt modified	driptanil
readme					hello.txt file added	driptanil
REMOTE 3/3						
origin						
master						
new-feature						
readme						

```

driptanil@driptanil in repo: LearningGit on ❷ not-feature took 3ms
λ git branch -r
origin/master
origin/new-feature
origin/readme

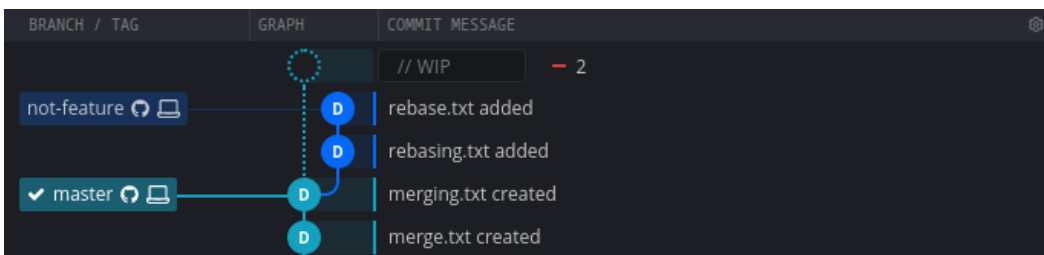
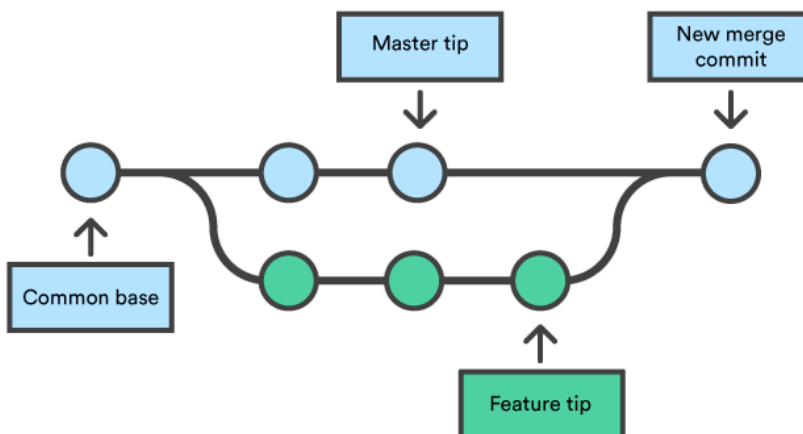
driptanil@driptanil in repo: LearningGit on ❷ not-feature took 23ms
λ git push origin --delete new-feature
To github.com:driptanil/LearningGit
- [deleted]          new-feature

driptanil@driptanil in repo: LearningGit on ❷ not-feature took 3s
λ git branch -r
origin/master
origin/readme

```

## Merging Branch

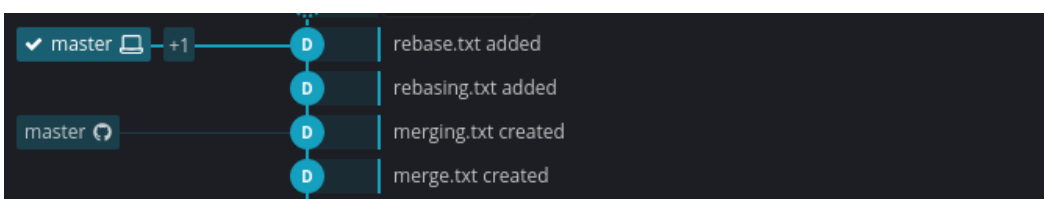
- `git merge <branch>` integrates the changes of the branch in the local head branch.



```

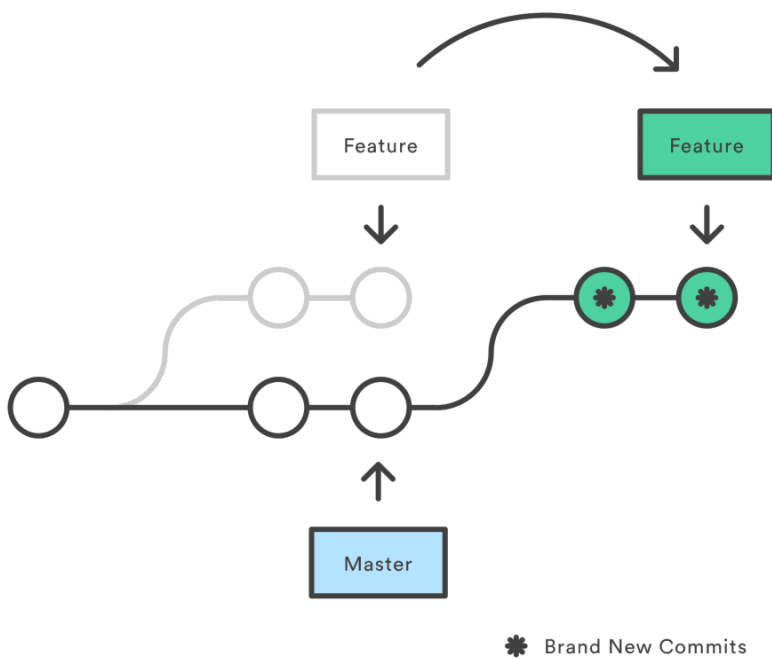
driptanil@driptanil in repo: LearningGit on ❷ master [x] took 4ms
λ git merge not-feature
Updating 85f5c05..af4eb47
Fast-forward
 rebase.txt | 0
 rebasing.txt | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 rebase.txt
 create mode 100644 rebasing.txt

```



## Rebasing Branch

- `git rebase <branch>` integrates the changes of branch with the local head branch.



BRANCH / TAG	GRAPH	COMMIT MESSAGE
master	D	merging.txt created
	D	merge.txt created
✓ not-feature	D	Subject: Renamed status.txt to grattude.txt
master	D	name.txt created
readme	D	Subject: Renamed status.txt to grattude.txt
	D	hello.txt
	D	hello.txt modified
	D	hello.txt file added

```

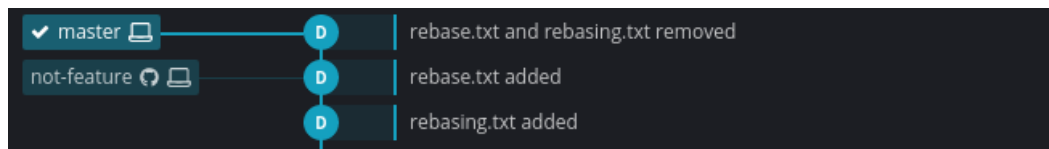
driptanil@driptanil in repo: LearningGit on  master [!2] took 20ms
λ git rebase not-feature
Successfully rebased and updated refs/heads/master.

```

BRANCH / TAG	GRAPH	COMMIT MESSAGE
✓ master	D	merging.txt created
	D	merge.txt created
not-feature	D	Subject: Renamed status.txt to grattude.txt
master	D	name.txt created
readme	D	Subject: Renamed status.txt to grattude.txt
	D	hello.txt
	D	hello.txt modified
	D	hello.txt file added

## Comparing Branches

- `git log <branch>...<another_branch>` displays the commits which are made in a branch but not in other branch.



```
driptanil@driptanil in repo: LearningGit on  $\wp$  master [?3] took 13ms
λ git log not-feature..master
commit da7b882904ad62157032dd88c9bd321dddbc465 (HEAD → master)
Author: Driptanil <driprecovery@gmail.com>
Date: Wed Mar 30 02:14:45 2022 +0530

    rebase.txt and rebasing.txt removed
```