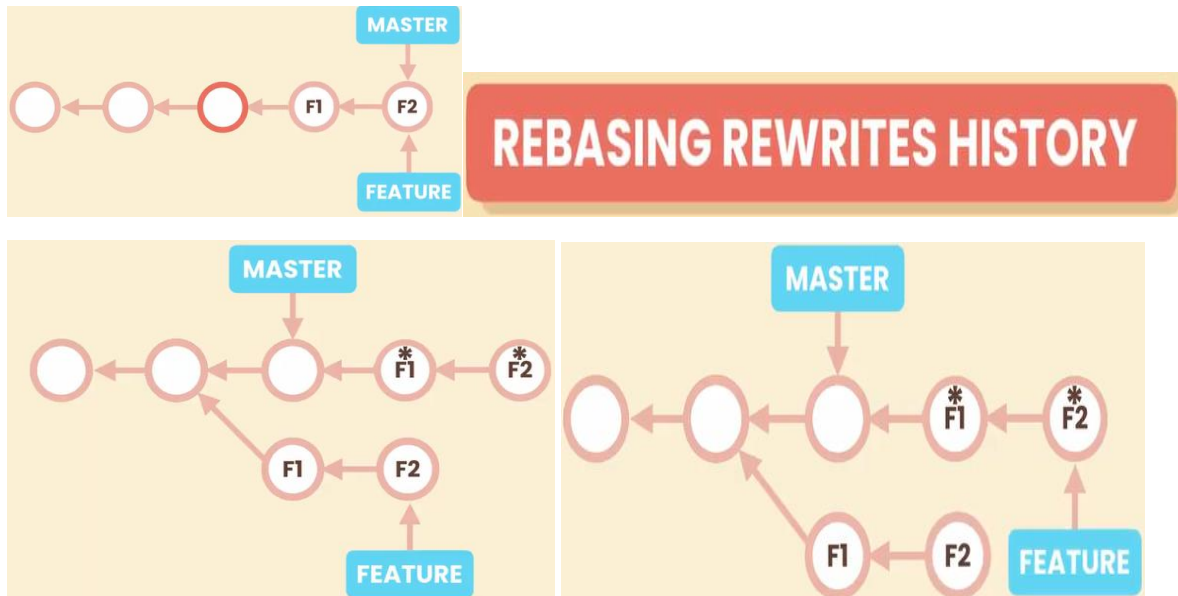


Rebasing:

For Merging, We Will End With Fast-Forward Merging.

We Must Use Rebasing For Local Repository (That Means With Re-Basing Not Use With Push).

Here In The Third Picture, Rebasing Will Create New Commits For F1 & F2 And Change The Pointer.



To Rebase:

- First: Go To The Specific Branch.
- Run Command: git rebase main

```
Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ git switch feature/shopping-card
Switched to branch 'feature/shopping-card'

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (feature/shopping-card)
$ git rebase main
Successfully rebased and updated refs/heads/feature/shopping-card.
```

```
Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (feature/shopping-card)
$ git log --oneline --all --graph
* a5a1ed5 (HEAD -> feature/shopping-card) Adding Cart.txt
* e595bff (main) Update file-01.txt
* ba352bf Fix The Bug On The Photo Upload Page.
* 6969780 Merge branch 'bug-fixing-03' Into Main Branch.
| \
| * 758d0b3 (bug-fixing-03) Changes The File-05.txt To Test Merging
| /
* df13941 (bug-fixing-02) Adding New File For Fast-Forward Merging
* 076b09b Compare Changes For File-03
| \
| * 048e6e4 Adding New Line For File-04
| * 10b7bed Test Branches For Bug Fixing
* | 9df8a5c Adding File-03 With Changes To Main Branch
| /
| * 2d80b85 (refs/stash) On main: For Testing The VSCode With Branches
| /|
| * 0747d0d index on main: a6f4fa3 Restore File After Deleting
| /
* a6f4fa3 (tag: V1.1) Restore File After Deleting
* 715de30 Deleting The New File: file-03.txt
* 1b6514e Adding All Files
* 43a35dd Second Testing Of Files To Show Output
* c5b550f Testing Short Status Command
* edc6933 Removing File From Index/Staging Area
* c047d34 Adding By Accident
* 014c4b9 Adding .gitignore To Project
* 2df21af Trying Rename Files
* 4702e82 After Deleting File2.txt
* c378284 Initial Commit
```

```
Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ git merge feature/shopping-card
Updating e595bff..a5a1ed5
Fast-forward
 cart.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 cart.txt
```

If We Have Conflict With Rebasing, Then We Must Solve The Conflict, Then Use Command: **git rebase --continue**

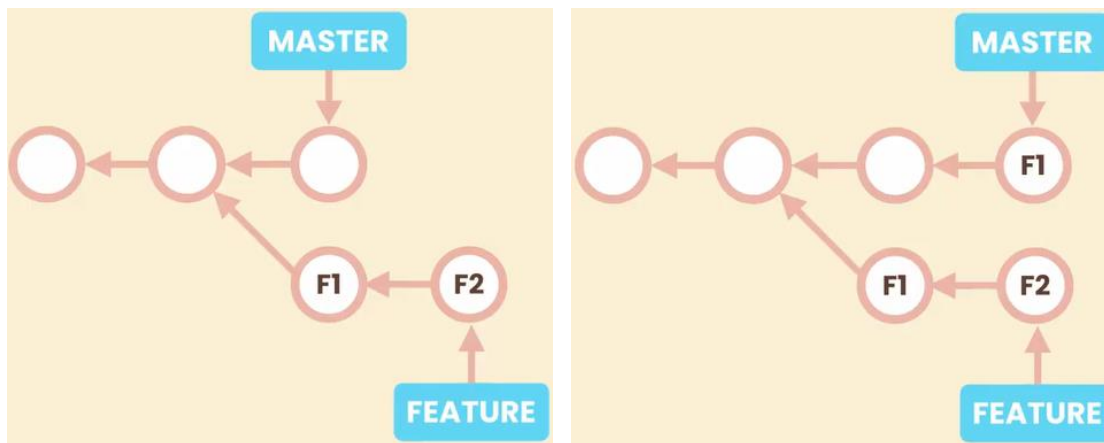
```
~/Projects/Venus git feature/shopping-card | rebase-i
> ✓ git rebase --continue
```

```
~/Projects/Venus git feature/shopping-card | rebase-i
> ✓ git rebase --skip
```

```
~/Projects/Venus git ♀ feature/shopping-cart | rebase-i
> ✓ git rebase --abort
```

```
~/Projects/Venus git ♀ feature/shopping-cart
> ✓ git config --global mergetool.keepBackup false
```

Cherry Picking: We Have Feature In Some Branch, And We Want It On Master.



Here We Use The Commit-Id Of The Feature That We Want On Master (Main) Branch

```
~/Projects/Venus git ♀ master
> ✓ git cherry-pick 5670ecc
Auto-merging toc.txt
CONFLICT (content): Merge conflict in toc.txt
error: could not apply 5670ecc... Write mountain to toc
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'

~/Projects/Venus git ♀ master | cherry
> 1 git mergetool
Merging:
toc.txt

Normal merge conflict for 'toc.txt':
{local}: modified file
{remote}: modified file
```

To Restore Specific File From Specific Branch (Not The Entire Feature); We Can Use: *git restore*

- *Ex: git restore --source=feature/send-email --file-01.txt*

```
Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ git merge feature/shopping-card
Updating e595bff..a5a1ed5
Fast-forward
 cart.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 cart.txt

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ git switch -C feature/send-email
Switched to a new branch 'feature/send-email'

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (feature/send-email)
$ echo river >> file-0
File-05.txt  file-01.txt  file-02.txt  file-03.txt  file-04.txt

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (feature/send-email)
$ echo river >> file-01.txt

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (feature/send-email)
$ git commit -am "Update File-01.txt"
warning: in the working copy of 'file-01.txt', LF will be replaced by CRLF the next time Git touches it
[feature/send-email aaca04f] Update File-01.txt
 1 file changed, 1 insertion(+)

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (feature/send-email)
$ git switch main
Switched to branch 'main'

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ git restore --source=feature/send-email file-01.txt

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ cat file-01.txt
hello
world
Jafar Loka
Jafar Loka Here
Jafar Loka -02
bugfix
hello-world-02
river
```

```
Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ git remote add origin https://github.com/gaaferHajji2/j-1-test-git.git
error: remote origin already exists.

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ ^C

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ git remote set-url origin https://github.com/gaaferHajji2/j-1-test-git.git
```

```

~/Projects
> git clone https://github.com/codewithmosh/Mars.git MarsProject
Cloning into 'MarsProject'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 589 bytes | 589.00 KiB/s, done.

```

To Get List Of Remote Repositories:

```

~/Projects/MarsProject master
> git remote
origin

```

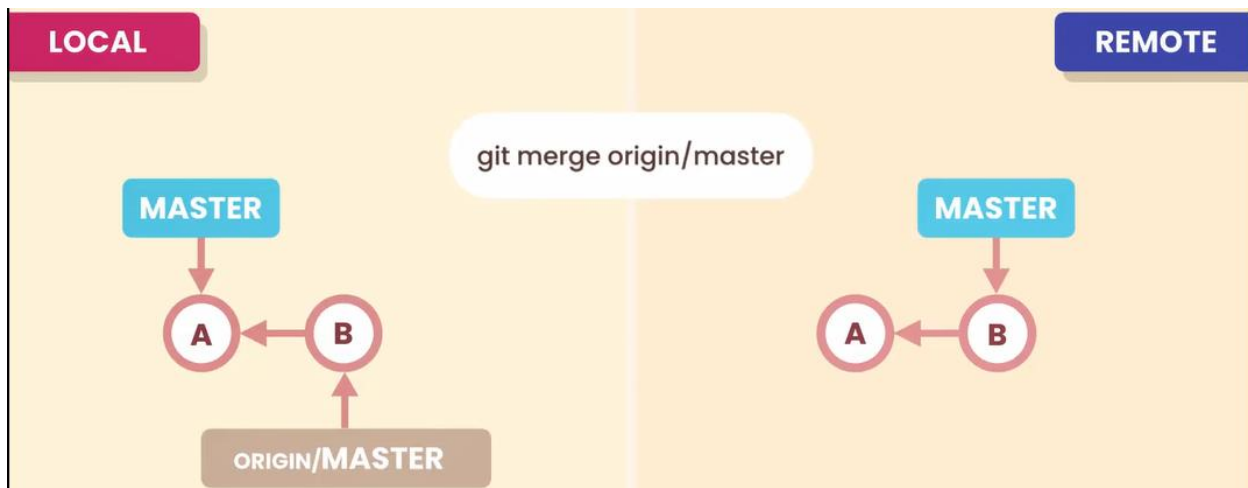
```

Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ git remote
origin
Tatweer@DESKTOP-9FV4RTN MINGW64 /c/Tests/Github-And-GitLab/Tutorial-02-Code-With-Mosh (main)
$ git remote -v
origin https://github.com/gaaferHajji2/j-1-test-git.git (fetch)
origin https://github.com/gaaferHajji2/j-1-test-git.git (push)

```

Here We Have Changes On Remote, But Not On Local; So We Use: *git fetch*.





If We Don't Set The Third Parameter For Fetch; *Then The Default is Origin.*

If We Don't Set The Fourth Parameter For Fetch, *Then The Default Will Be main.*

```
~/Projects/MarsProject > git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 647 bytes | 647.00 KiB/s, done.
From https://github.com/codewithmosh/Mars
 fb3b343..012ff85  master    -> origin/master
```

```
~/Projects/MarsProject > git log --oneline --all --graph
* 012ff85 (origin/master, origin/HEAD) Update README.md
* fb3b343 (HEAD -> master) Initial commit
```

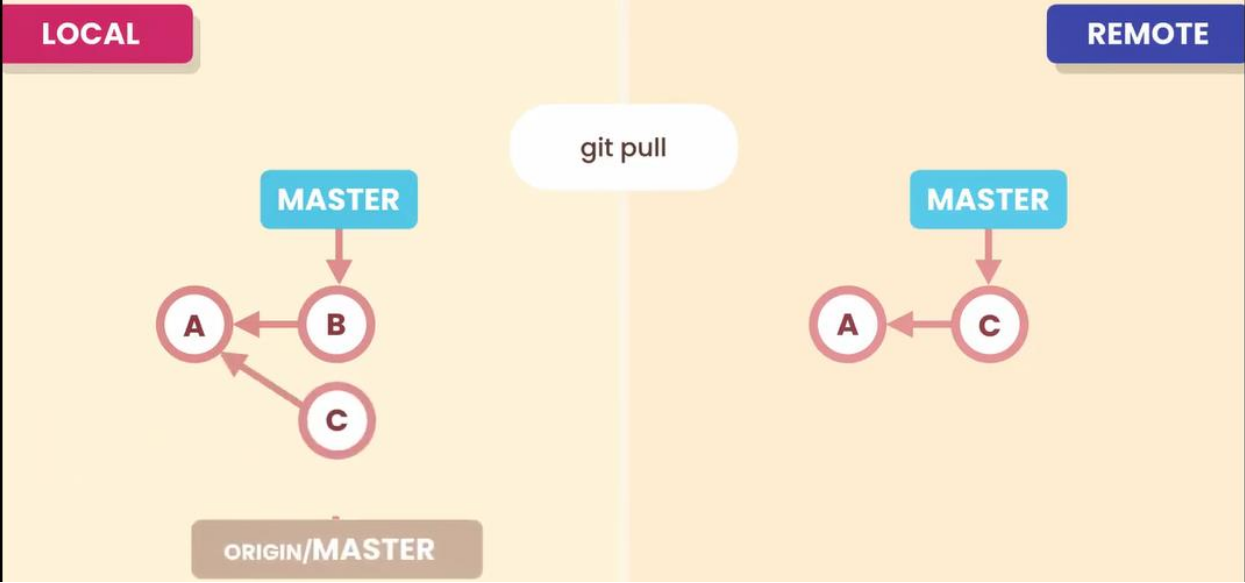
For See How The Local And Remote Branches Are Diverged

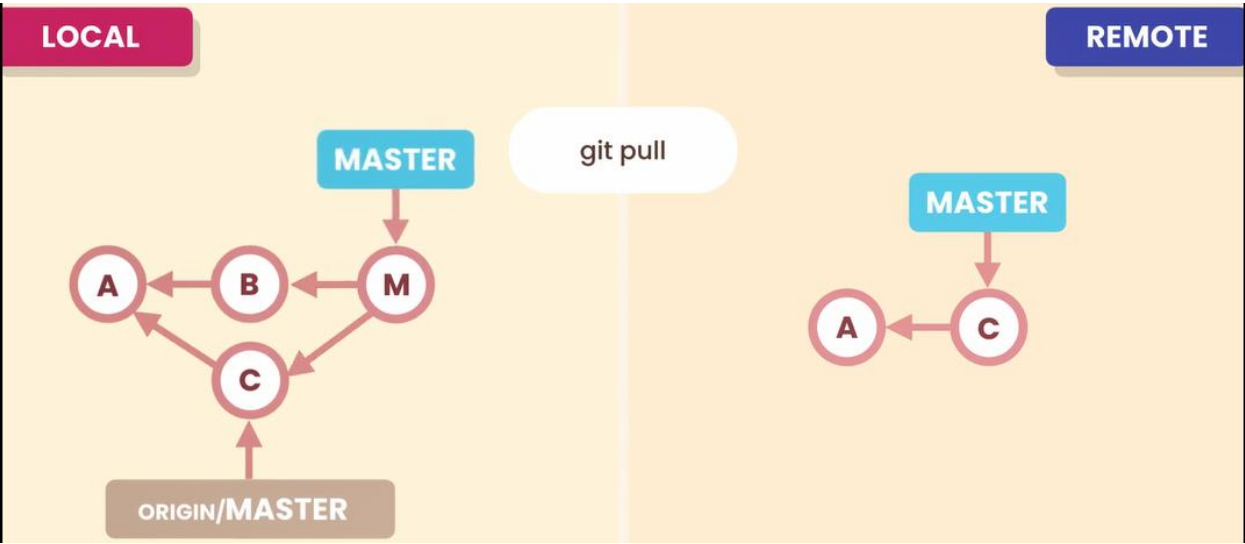
```
~/Projects/MarsProject > git branch -vv
* master fb3b343 [origin/master: behind 1] Initial commit
```

```
~/Projects/MarsProject master 1
> git merge origin/master
Updating fb3b343..012ff85
Fast-forward
 README.md | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)
```

pull = fetch + merge









Never Use This Command:



This Will Store Credentials For 50-Minutes:

```
~/Projects/MarsProject master  
> git config --global credential.helper cache
```

CREDENTIAL HELPER

- Mac: Keychain
- Windows: Windows Credential Store
