

- * GIT is a version control system (VCS)

```
PS C:\xampp\htdocs\SYGP> git config user.name
Rumindu De Silva
PS C:\xampp\htdocs\SYGP> git config user.email
rumindukavishka14@gmail.com      --global
PS C:\xampp\htdocs\SYGP> git config user.name "Rumindu"
PS C:\xampp\htdocs\SYGP> git config user.name
Rumindu
PS C:\xampp\htdocs\SYGP> git config user.name "Rumindu De Silva"
PS C:\xampp\htdocs\SYGP> git config user.name
Rumindu De Silva
```

like above we can change email also.

- * One folder only contain one git repo. (but nested folders doesn't need initialise another repo)
- * Before done any thing on git we should create git repository. using git init
- * Without creating a git repo if we just type git status it will show

```
PS C:\xampp\htdocs> git status
fatal: not a git repository (or any of the parent directories): .git
```

- * "git init" run only one time in a project .
- * After run "git init", ".git" file create . It's a hidden file. it will store whole the info about version control history .

```
rumin@Rumindu MINGW64 /c/xampp/htdocs/SYGP (main)
$ rm -rf .git          // Deleting .git file using git bash .
rumin@Rumindu MINGW64 /c/xampp/htdocs/SYGP
$ git status
fatal: not a git repository (or any of the parent directories): .git
```

- * If you delete .git whole project will delete

- * Git tracks a directory and all nested subdirectories
But not tracking parent directory.



- * One repo per project .
- * Commit :- checkpoint or snapshot. of a repository .
- * Git repo has many commits .
- * There is a message along with commit . This message should summarize changes that are included in commit .

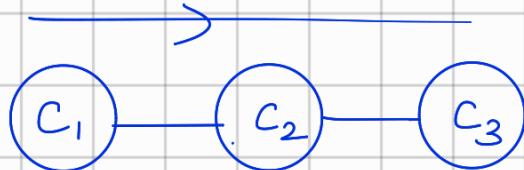
↓ Add /Delete /update
- * Before the commit we need to add these file in to staged area using git add <file name>
git add . :- to stage all changes at once .
- * After file /files to staging area we can commit the file using git commit -m "commit message" or git commit :- Here will open default git text editor and you have to type commit message
- * git commit -am "commit message" :- Add & commit ⁱⁿ single code

* Atomic Commit :- keep each commit focused on a single thing
Here just add relevant files in to staging area using `git add <file names>` and commit them.

* git commit message :- Present tense & imperative style. (according to git docs)

ex:- make xyz do fro + 2 -

* Ammending commits :- Just immediately done a commit .
We can't amend a commit if there is a commit after an ammending commit .



- Here we can only amend C3 only -

`git commit --amend` :- amend commit message .

```
> git commit -m 'some commit'  
> git add forgotten_file  
> git commit --amend
```

If some file is forgotten to add to commit .

* `.gitignore` :- Google what are files contain in `.gitignore` in React project and add those files .

* `git log --oneline` :- Show all commit history within one line .

Branching

- * The default branch name is master. It doesn't have fancy powers. Its just like any other branch.
- + Git hub's default branch name is main.

```
commit 9c5a589b8a061815f05527c324bcd7586187b11 (HEAD -> master) — ?  
Author: Rumindu De Silva <rumindukavishka14@gmail.com>  
Date: Thu Jun 29 17:05:21 2023 +0530
```

- * Head is what is currently we seen.
- * git branch :- to see all branches in the repo . "*" active branch
- * git branch <branch-name> :- Just creates the branch, doesn't switch to that branch. (the Head stays the same).
- * git switch <branch-name> :- Switching to already created branch (head is moving to this branch)
- * git switch -c <branch-name> :- Creating a branch and switching to it.
- * git checkout <branch-name> :- like switch command but this is older one . switching to existing branch.

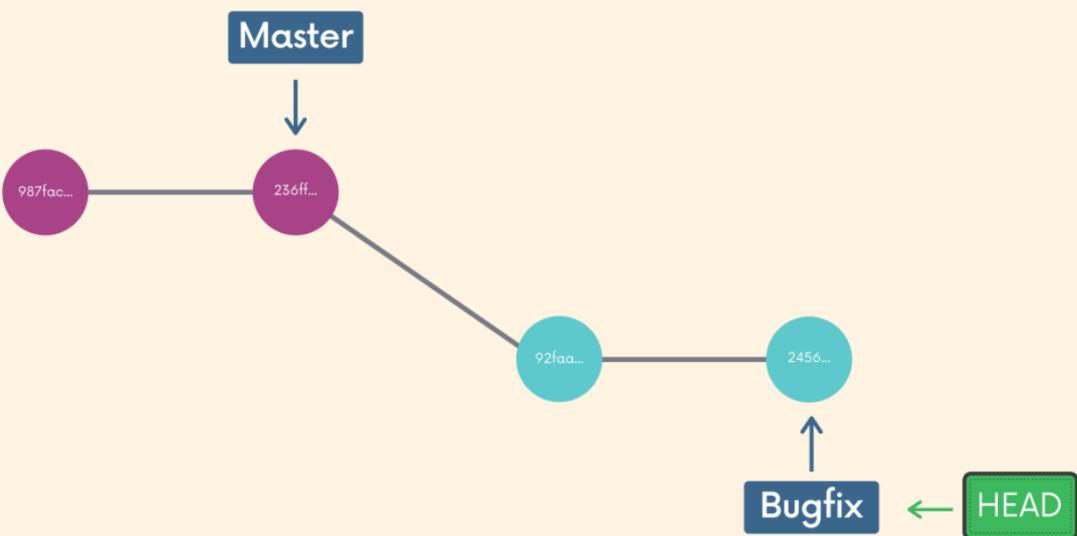
- * `git branch -d <branch-name>`: Delete the branch. But we can't delete the branch while on the branch.
- * `git branch -m <new branch name>`: Rename the branch. For Rename branch first we need to switch relevant

Merging

Fast forward merging :- No need to resolve merge conflict these types of mergings.

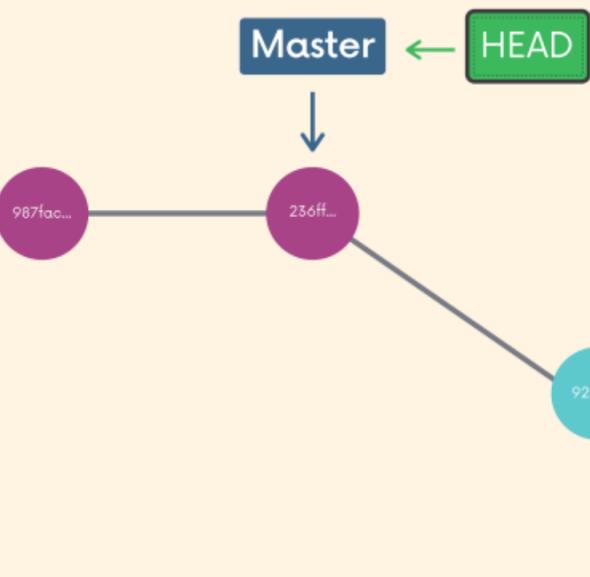
* We need to merge Bugfix branch to Master branch.

To merge the bugfix branch
into the master branch...



* At very first we need to switch master branch.

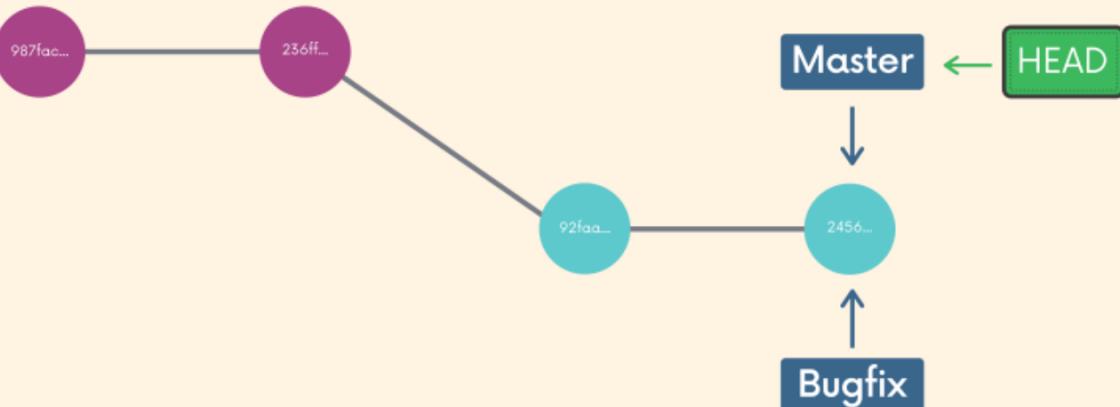
git switch master



Switch to the master branch

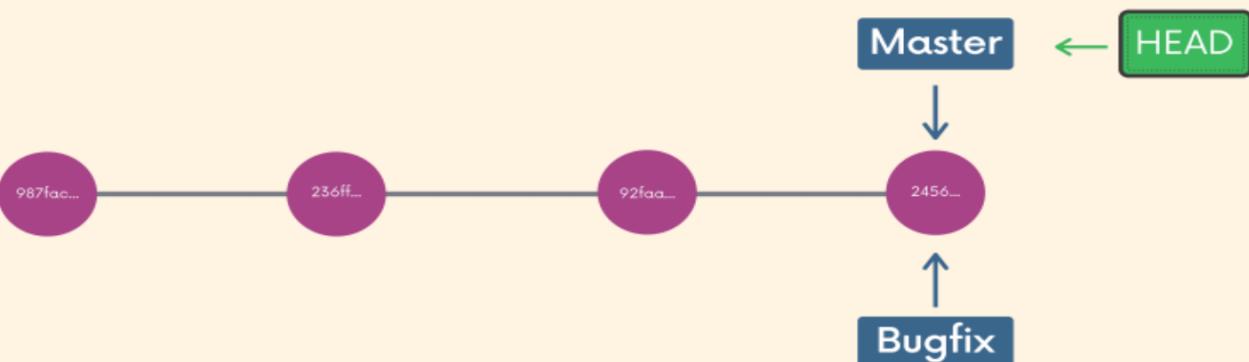
git merge bugfix

Merge bugfix into master



This Is Called A Fast-Forward

Master simply caught up on the commits from Bugfix



Git stash

When I try to switch branch while having uncommitted changes (to see some stuff in branches)

- 1) My changes come with me to the destination branch (If there conflict isn't conflict)
- 2) Git won't let me switch if it detects potential conflicts (use git stash)

git stash :- put uncommitted changes in to stash

git stash pop :- get back most recent stash in to current branch.