GIT

Git is free and open source version control system for managing files and tracking change among multiple people. It is created by Linus Torvalds in 2005 for development of the Linux kernel.

A Git repository, encompasses the entire collection of files and folders associated with a project, along with each file's revision history by the developers of the project which helps in tracking the project and collaboration. A commit in a git repository records a snapshot of all the files in your directory and exist in linked-list relationship and can be organized into multiple lines of development called branches.

INSTALLATION PROCEDURE OF GIT

In Windows:

1. Git can be downloaded from Git windows installer page
2. Run the downloaded executable file
3. Open the Git bash shell
4. Configure the git with email id and username with the commands
   $git config --global user.name "username"
   $git config --global user.email "emailid"

In Linux:

1. Git can be directly installed using apt-get command
   $sudo apt-get install git
2. To check whether the git is installed properly we can run the following command
   $git –version
3. Configure the git with email id and username with the commands
   $git config --global user.name "username"
   $git config --global user.email "emailid"

In Mac OS:

1. Git can be downloaded from Git MAC installer page
2. Run the downloaded executable file
3. Configure the git with email id and username with the following commands in the command line
   $git config --global user.name "username"

$git config --global user.email "emailid"

GIT COMMANDS

INTIALIZATION OF GIT

git init:

Create an empty Git repository or reinitialize an existing one

git clone:

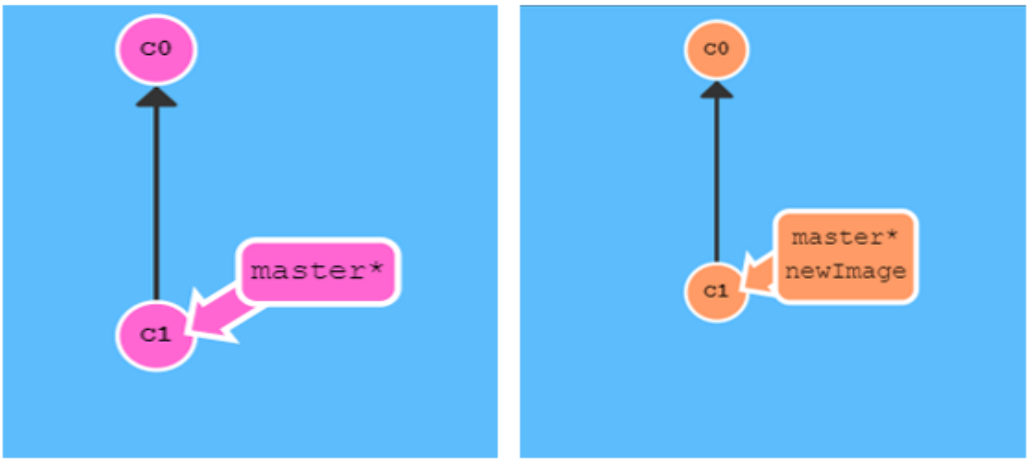
It creates a copy of an existing repository. It initially calls git init command and then copies all the files to the newly created repository

BRANCHING IN GIT

git branch branch_name:

A branch in Git is simply a lightweight movable pointer to one of the commit.

Ex: git branch newImage



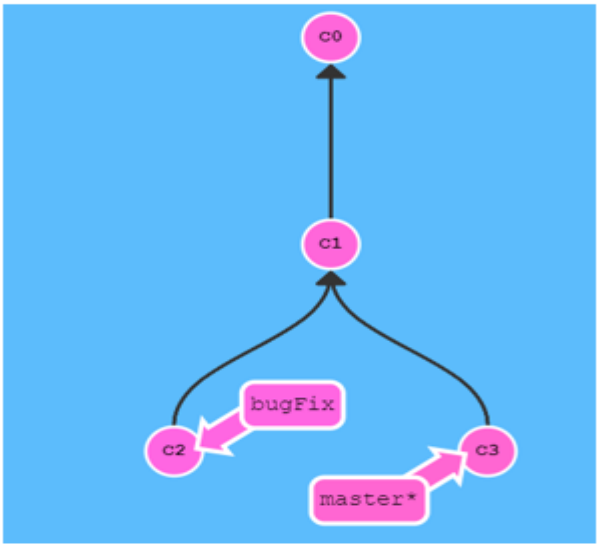
git branch -f branch1 branch2

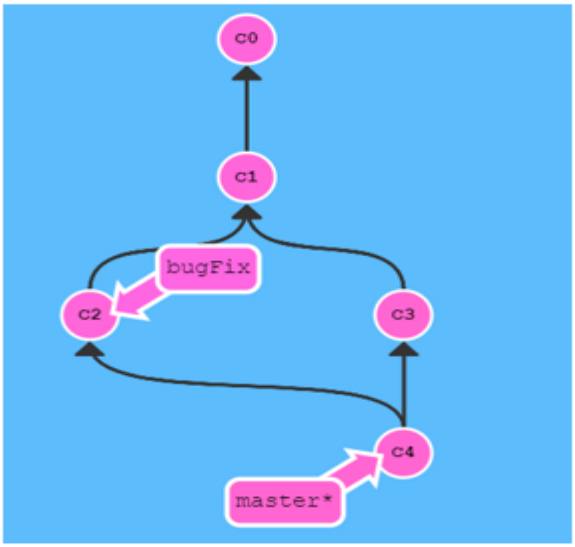
It moves the branch1 forcefully to the point pointed by branch2

git merge branch_name

Merging in Git creates a special commit that has two unique parents.

Ex: git merge bugFix



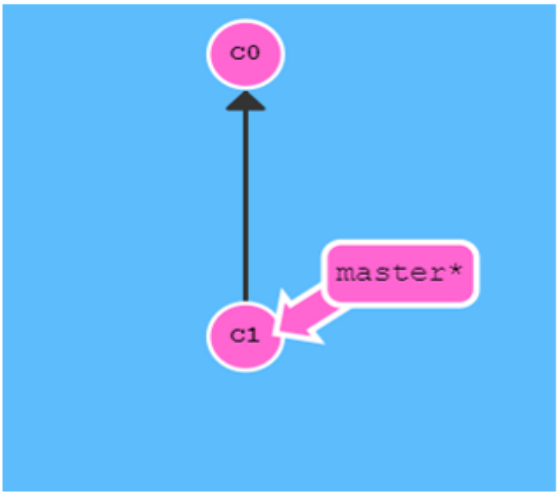


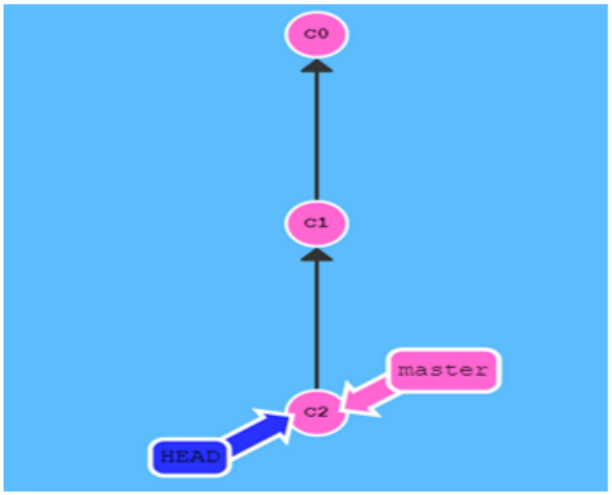
CHECKOUT IN GIT

git checkout branch_name

HEAD always points to the most recent commit which is reflected in the working tree. With the help of the checkout command, we can move the direction of the head to the commit that we wanted it to point to.

Ex: git checkout C2





git checkout -b branch_name:

It creates a new branch “branch_name” and the HEAD pointer points to the newly created branch.

git checkout branch_name^:

It points to the immediate parent node that is to the top of branch_name

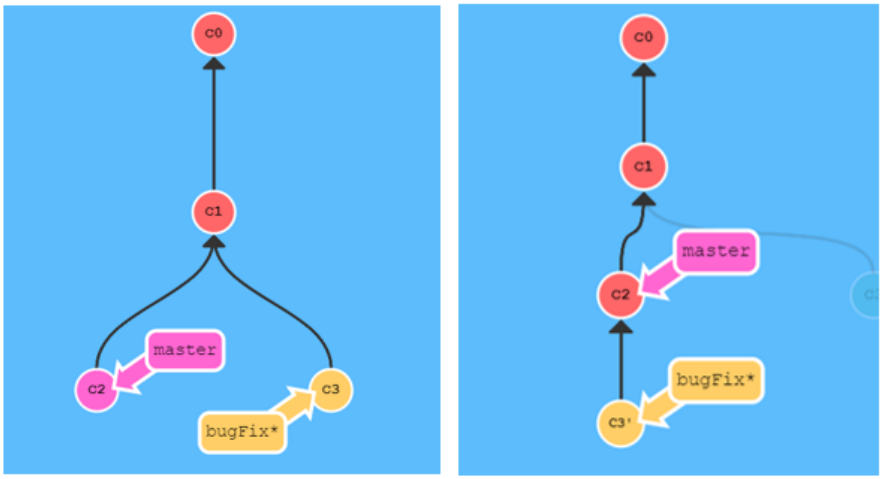
git checkout branch_name¬number:

It points to the parent node that is "number" levels to the top of branch_name

git rebase branch_name

Rebasing essentially takes a set of commits, "copies" them, and plops them down somewhere else.

Ex: git rebase master



OTHER USEFUL COMMANDS IN GIT

git status

The git status command displays the state of the working directory, the staging area and the unstaged area

git add filename:

It adds the file "filename" to the current content found in the working directory

git diff:

It adds the file "filename" to the current content found in the working directory

git reset:

Reset current HEAD to the specified state

git revert HEAD¬NUMBER

Revert the changes specified by the “number” last commit in HEAD and create a new commit with the reverted changes.

git push:

To push commits made on your local branch to a remote repository.

git pull:

Incorporates changes from a remote repository into the current branch.