

# Git Info

- Git is a distributed version control system to track the changes in source code.
- Working tree - currently working.
  - ↳ Contains files NOT handled by Git (also known as untracked files).
  - ↳ When you make changes to your files they are made in working tree
- Staging area - Index
  - ↳ When you do `git add` command, Git takes a snapshot of the changes you've made in your files & adds this snapshot to staging area
  - ↳ Staging area contains files that are going to be a part of the next commit.
- Commit - Checkpoint
  - ↳ Tells Git to track all changes that have occurred up to this point using the last commit as a comparison
  - ↳ Git records all changes in staging area & puts them in local repo. After committing, staging area will be empty



## Delete a file & Stage

↳ `rm example.txt`

↳ Stage the deletion: you need to tell Git to stage this deletion

↳ `git add example.txt`

git add is used to stage any changes to tracked files, not just additions. This includes any modifications or deletions.

"I want to include this deletion in my next commit"

↳ `git rm example.txt`

↳ Commit the change: `git commit -m "Delete example.txt"`



↳ `git rm --cached` file removes a file from Git index (staging area) but not from working directory



## Modify a Commit

• `git commit --amend` → takes your staging area & uses it for a new commit. If you've made no changes since your last commit, then your snapshot stays same & all you'll change is commit message

## Change the Commit message

## Branches

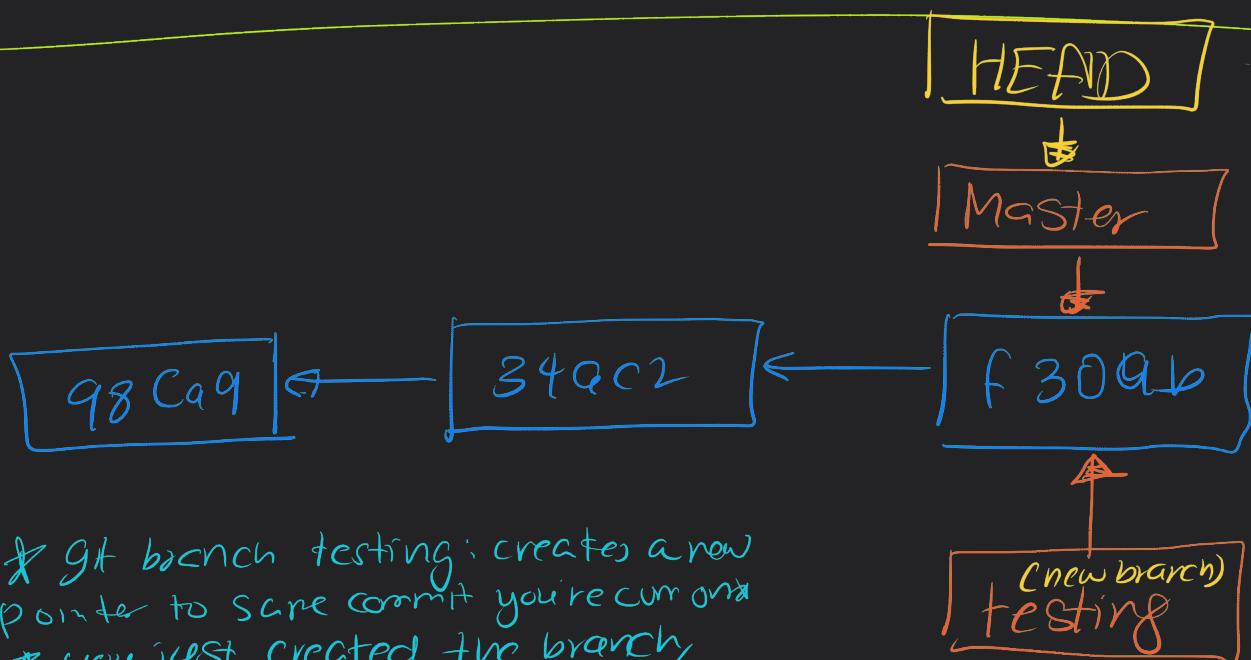
- A branch is essentially a unique series of code changes with a unique name.

git branch new-branch

git checkout new-branch

1) Pointer to a commit: A branch can be thought of as a pointer to a commit. When you create a new branch, it points to the same commit that you're currently on.

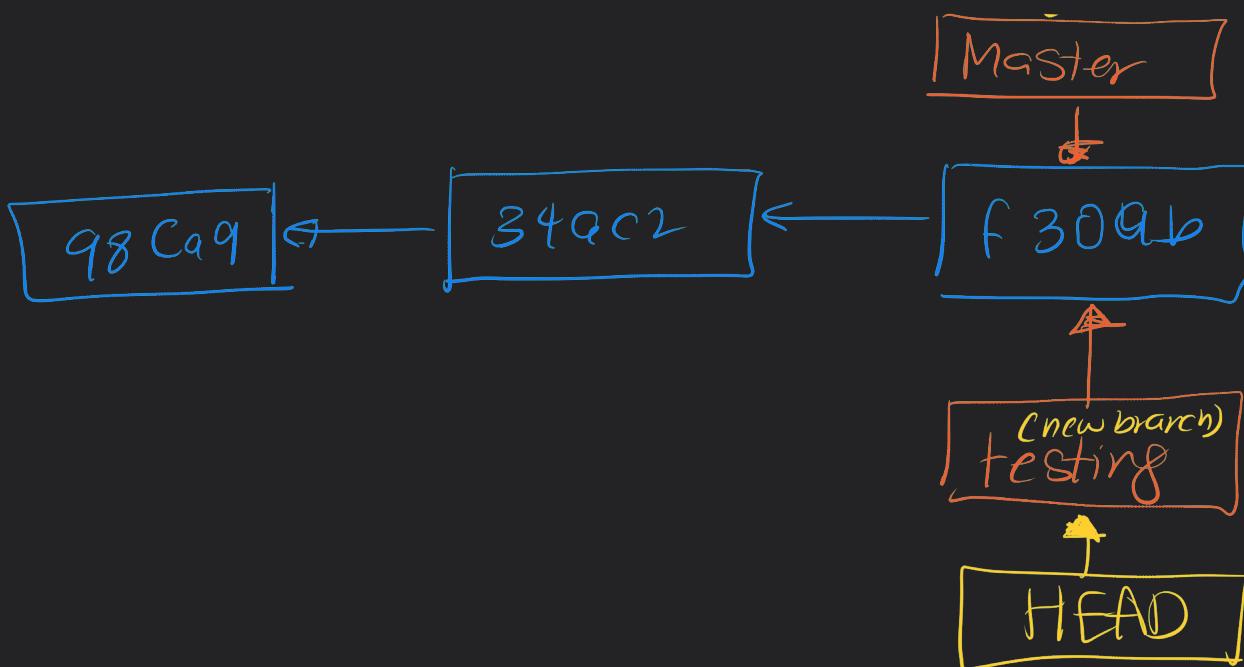
→ Head pointer: special pointer to know which branch you're on



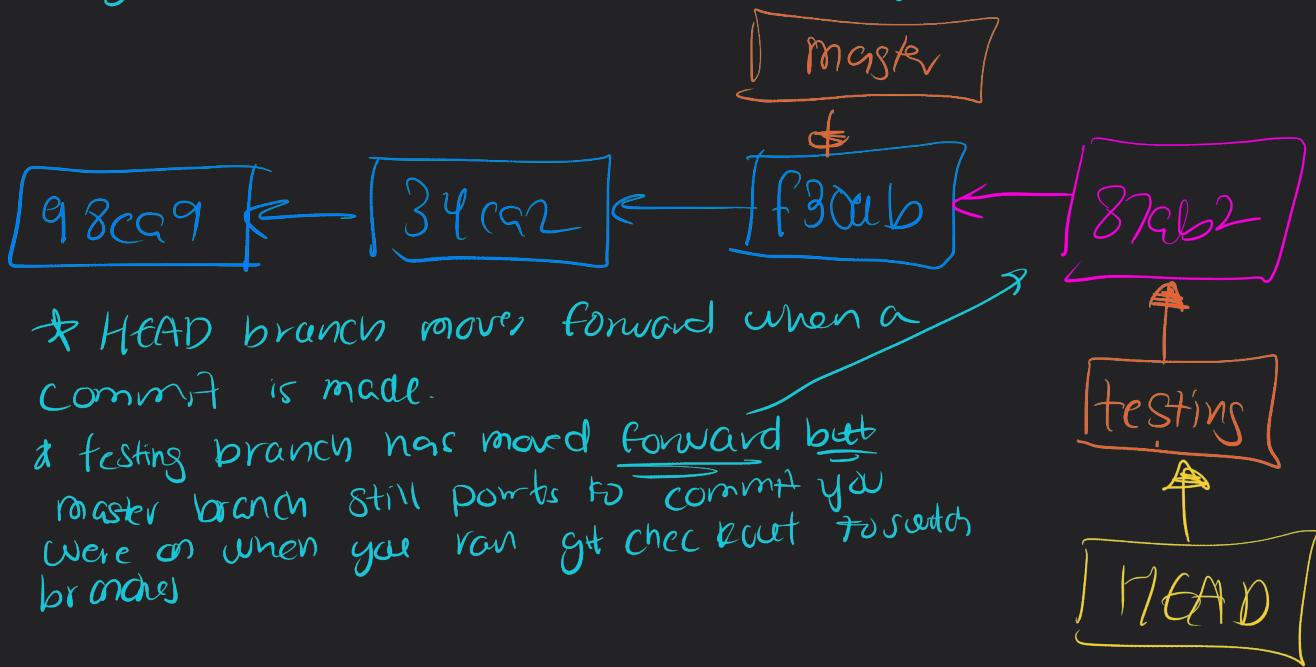
\* git branch testing: creates a new pointer to same commit you're current on  
\* you just created the branch, so HEAD still points to master.

\* git checkout testing - moves HEAD to point to testing branch

Current branch



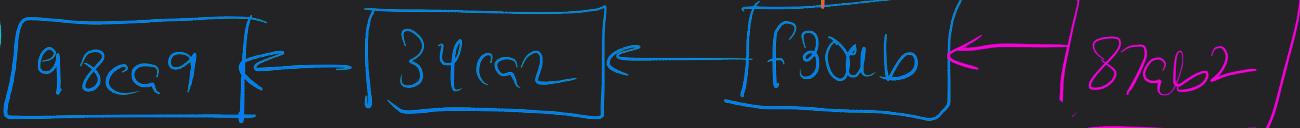
\* git commit -a -m "Make a change" -



\* HEAD branch moves forward when a commit is made.

\* testing branch has moved forward but master branch still points to commit you were on when you ran git checkout to switch branches

\* git checkout master  
\* Head moves when you checkout



→ also it reverted the files in your working directory back to the snapshot that master points to

\* git reset

undoing changes

git reset HEAD <file> command unstaged

the changes made to the specified file. This means that the change to that file WILL NOT be included in the next commit unless you add the file to the staging area again with git add

So how do I know where a branch starts and where it ends? Git Log

commit  
DRAFT  
branch  
repo abc  
branch

new  
branch  
diverged  
from  
main  
client  
one

- \* fcc756f (master) resolved merge conflict
- | \* d8d3c34 (gui) improved rendering for css
- | \* 02ff43b fixed rendering
- | \* 5f0e0d2 Merge branch 'cli'
- | \* | 48bb80d (HEAD, cli) added X11 timeout handling
- | \* | f0f6bbd added usage statement
- | \* | 109bc5f added command line argument parsing
- | / | 3a4c50c added readme file
- | / | merge -Cherry-pick from cli have  
| / | been merged back into master
- \* 5dc2d70 combined with web engine
- \* 3c14ad1 added stage and web view
- \* 2787158 boilerplate for graphical user interface
- \* 750a58f initial scaffolding for GitDawg web browser

scratches here  
Part of master and gui branch ends here  
unique to gui branch  
cl diverges from master/starts here  
cli ends part of cli  
current branch  
specific to cli branch  
part of master, cli, & gui  
oldest commit

- 1) oldest commit overall: 750958f ✓
  - 2) oldest commit in master's branch entire history 750958f ✓
  - 3) oldest commit in CLI branch's ~~109bc5f~~ 750958f ✓
  - 4) newest commit in git branch's entire history d8d3c34 ✓
  - 5) newest commit on current branch 498bb80d ✓
  - 6) newest commit overall fcc756f ✓

- `l` = continuation of branch
  - `*` = commit
  - `ll` = new branch has diverged from current point

