explain to kid

GIT

programming concepts

lets say you are coding
and your code is working fine

and you
close your
laptop →

next day, you mess up your code and it **stops working**

and now you wish that you can go back in time when it was working fine

tk in
me

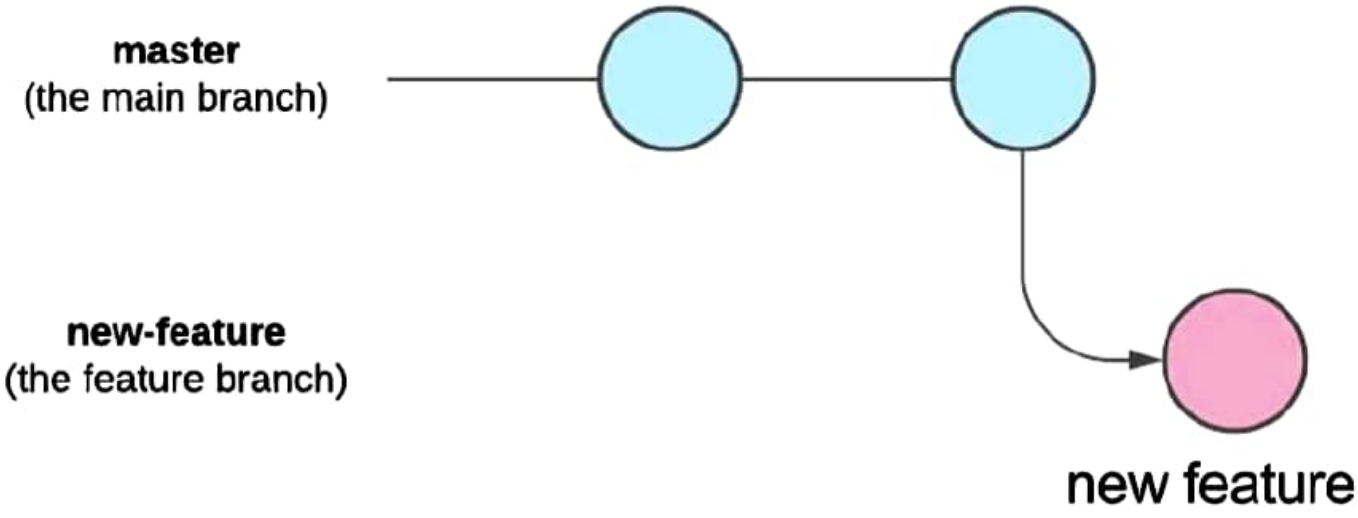
this is where we use **GIT**

Git is a tool that helps Coders track changes in their code

it does a lot more than that →

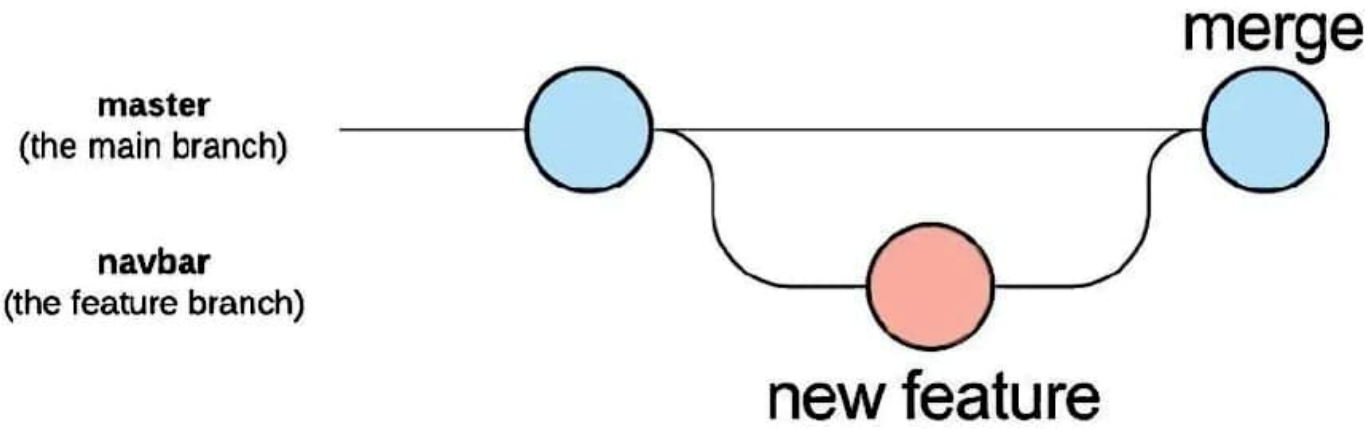
you can add a new feature by creating a new branch

anything you code in the new branch won't effect the main branch



and when you are happy with your new feature

you can merge your new branch with the main branch



ALL GIT COMMANDS

git init: Initializes a new Git repository.
git clone: Copies an existing repository into a new directory.
git add: Stages changes (files) to be committed.
git commit: Records changes to the repository with a message.
git status: Displays the state of the working directory and staging area.
git push: Uploads local repository content to a remote repository.
git pull: Fetches and merges changes from a remote repository into the current branch.
git fetch: Downloads objects and refs from another repository.
git merge: Combines two or more development histories together.
git branch: Lists, creates, or deletes branches.
git checkout: Switches branches or restores working tree files.

git commit --amend: Modifies the most recent commit.
git config: Sets configuration options for Git repositories.
git rev-parse: Parses revision (commit) identifiers.
git describe: Shows the most recent tag that is reachable from a commit.
git shortlog: Summarizes git log output by author and commit message.
git grep: Searches for patterns in tracked files.
git gc: Optimizes the repository by cleaning up unnecessary files and compressing file history.
git fsck: Verifies the integrity of the repository.
git commit -a: Stages all modified and deleted files and commits them.
git remote add: Adds a new remote repository.

git log: Shows the commit history for the repository.
git reset: Undoes changes by resetting the index and/or working directory.
git rebase: Reapplies commits on top of another base tip.
git diff: Shows the changes between commits, commit and working tree, etc.
git remote: Manages remote repository connections.
git tag: Creates, lists, deletes, or verifies tags.
git stash: Temporarily saves changes in a "stash" and reverts the working directory to the last commit.
git stash pop: Applies stashed changes and removes them from the stash list.
git stash apply: Applies stashed changes without removing them from the stash list.
git cherry-pick: Applies the changes introduced by an existing commit.

git rm: Removes files from the working directory and staging area.
git mv: Renames or moves a file, directory, or symlink.
git show: Displays information about a git object, such as a commit, tree, or blob.
git blame: Shows what revision and author last modified each line of a file.
git clean: Removes untracked files from the working directory.
git bisect: Finds the commit that introduced a bug by binary search.
git reflog: Shows a log of references for the local repository.
git archive: Creates a tar or zip archive of the repository.
git submodule: Manages submodules within a repository.
git revert: Reverts a previous commit by creating a new commit.

git remote remove: Removes a remote repository.
git merge --no-ff: Merges branches and creates a merge commit even if a fast-forward merge is possible.
git pull --rebase: Fetches the remote content and replays local commits on top of it.
git push --force: Forces an update to a remote branch, potentially overwriting history.
git tag -a: Creates an annotated tag in Git.
git format-patch: Prepares patches for e-mail submission.
git apply: Applies a patch to files and/or to the index.
git cherry: Lists commits that are in one branch but not another.
git ls-files: Shows information about files in the index and the working tree.