

Deleting Branches

- ❖ When you are done with a branch and it is no longer needed then you can delete the branch
 - `git branch -d contact-form`
- ❖ Deleting remote branch, add "r" flag
 - `git branch -dr origin/contact-form`

Undoing Local Changes

- ❖ If you have local changes that are not committed and want discard change then you can use following commands
- ❖ Discard changes in single file
 - `git checkout HEAD <file/to/restore>`
- ❖ Discard all changes that are not committed (already learned previously)
 - `git reset --hard HEAD`

Undoing Committed Changes

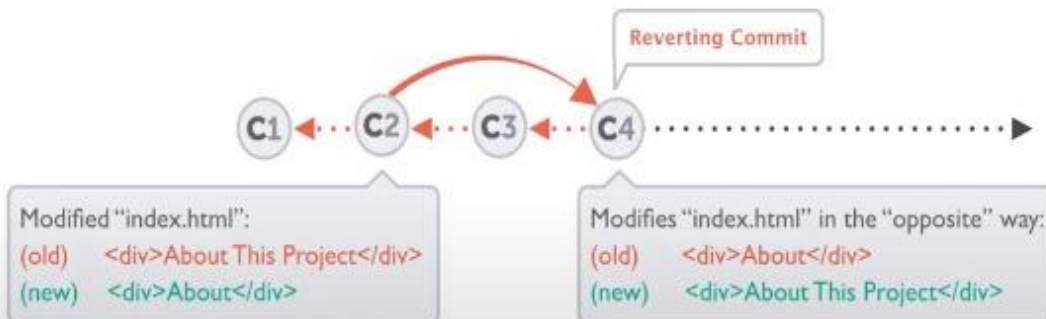
- ❖ Once code is committed and you feel your changes were wrong and you want to undo commits
- ❖ You can undo committed changes using following commands:
 - `git revert <commitHash>`
 - `git reset --hard <commitHash>`

`git revert <commitHash>`

- ❖ This command does not actually delete any commit.
- ❖ Instead it reverts the effects of a certain commit, effectively undoing it.
- ❖ It does this by producing a new commit with changes that revert each of the changes in that unwanted commit.
- ❖ For example, if your original commit added a word in a certain place, the reverting commit will remove exactly this word, again.

git revert <commitHash>

- ❖ For example: `git revert 2b504be`



git reset --hard <commitHash>

- ❖ It neither produces any new commits nor does it delete any old ones.
- ❖ It works by resetting your current HEAD branch to an older revision (also called "rolling back" to that older revision):
- ❖ If you call it with "--keep" instead of "--hard", all changes from rolled back revisions will be preserved as local changes in your working directory.

`git reset --hard <commitHash>`

- ❖ It neither produces any new commits nor does it delete any old ones.
- ❖ It works by resetting your current HEAD branch to an older revision (also called “rolling back” to that older revision):
- ❖ If you call it with “--keep” instead of “--hard”, all changes from rolled back revisions will be preserved as local changes in your working directory.

Work Flow

Github Flok

Remote

```
C:\Repo\Spoon-Knife>git remote -v
origin https://github.com/zeeshanhanif/Spoon-Knife.git
origin https://github.com/zeeshanhanif/Spoon-Knife.git

C:\Repo\Spoon-Knife>git remote show origin
* remote origin
```

```
origin https://github.com/zeeshanhanif/Spoon-Knife.git (fetch)
origin https://github.com/zeeshanhanif/Spoon-Knife.git (push)

C:\Repo\Spoon-Knife>git remote add upstream https://github.com/octocat/Spoon-Knife.git

C:\Repo\Spoon-Knife>git remote -v
origin https://github.com/zeeshanhanif/Spoon-Knife.git (fetch)
origin https://github.com/zeeshanhanif/Spoon-Knife.git (push)
upstream https://github.com/octocat/Spoon-Knife.git (fetch)
upstream https://github.com/octocat/Spoon-Knife.git (push)

C:\Repo\Spoon-Knife>git rem
```

Github pages

Create a branch **gh-pages** and deploy a static site