

Git Resetting and Reverting

- In this section, we will take a look at `git reset` and `git revert` command.
- Everyone makes a mistake, in some situation we don't want to commit certain things but do. So there is a several options to undo that commit.
- One of the option is `git revert` command. Second one is `git reset` command.

Git revert

- `git revert` command creates a new commit, which literally reverse the only changes that we made on the commit that we specified. A `git revert` command is useful if you want to undo the changes and keep those changes in your GIT history.

```
$ git revert <commit-id>
```

Git reset

- In the `git reset` command, there is two ways to reset the commit. Either with the `--soft` flag which we wants to keep the changes that we made or over the `--hard` flag in order to loose all the changes that we made on that commit.

```
# soft reset
$ git reset --soft HEAD~1

# hard reset
$ git reset --hard HEAD~1
```

- When we reset the commit with `--soft` flag, we still have the access to changes that we made by that commit. We can see that status by `git status` command. We can easily create an another commit this way.

```
$ git reset --soft HEAD~1
$ git status
On branch sarah
Changes to be committed:
  added:   third_story.txt           # file shown with git status command
```

- In the `--hard` flag, it will reset the commit without saving all those changes.

```
$ git reset --hard HEAD~1
$ git status
On branch sarah
Nothing to commit
```