* Notes
* Repository: **is a storage location for software packages.**
* Commits**: a commit is an operation which sends the latest changes of the source code to the repository**
* Continuous integration: **It's a primary DevOps best practice, allowing developers to frequently merge code changes into a central repository where builds and tests then run.**
* Git branching**: allows each developer to branch out from the original code base and isolate their work from others.**
* Notes
* Merging: **combining the work from two different branches together. This will allow us to branch off, develop a new feature, and then combine it back in.**
* Rebase**: Rebasing essentially takes a set of commits, "copies" them, and plops them down somewhere else.**
* Head: **points to the most recent commit which is reflected in the working tree. Most git commands which make changes to the working tree will start by changing HEAD.**
* Git commands
* Git commit: Make new commit
* Git branch (name): Create new branch.
* Git checkout (name): Select the branch.
* Git merge (name): Combining the work from two different branches together.
* Git rebase: (name):  Rebasing is **the process of moving or combining a sequence of commits to a new base commit**
* Git checkout (name)^: moves the branch backwards.
* Git checkout HEAD^: moves the head.
* Git commands
* Git checkout HEAD~4: moves the head 4 times.
* Git branch –f name HEAD~3: moves any branch by specifying a name for it.
* Git reset HEAD~1:  move a branch backwards as if the commit had never been made in the first place
* Git revert HEAD: Rewriting history" doesn't work for remote branches that others are using.
* Git commands
* Git cherry-pick c3 c4: copy a series of commits below your current location.
* Git rebase -i HEAD~3: keep all commits or drop specific ones.
* Git commit --amend: renaming commits.
* Git Tags: Mark certain commits as milestones, that you can reference like a branch.
* Git Clone: To create local copies of remote repositories(from github for example).
* Git commands
* Git fetch: downloads the commits that the remote has but are missing from our local repository, and updates where our remote branches point.
* Git pull: shorthand for a git fetch followed by a merge of whatever branch was just fetched.
* Git push: Publishing your work, it is responsible for uploading your changes to a specified remote. Once git push completes, all your friends can download your work from the remote.
* Git fakeTeamwork: