GIT

```
git clone <url>
```

- makes a copy of a repository
- stores it on your computer

- a "fork" creates your own copy of someone else's repository

---

```
git add
```

- adds a file to *staging area*
- tells `git` to include the file in the next revision to the repository
- `git add *` adds all changed files

```
git commit -m "message"
```

- saves the changes to repository as a new revision (a "commit")
- records a message
- `git commit -am "message"` adds and commits in same step

```
git status
```

- shows current status of repository

```
git push
```

- sends committed changes to remote repository
- more explicitly, could write `git push origin master`

```
git pull
```

- to syncronize the latest versions from remote repository to your own local machine

Merge Conflicts

- when two different commits can't be automatically merged

- need to be resolved

```
git log
```

- show a history of commits and messages

```
git reset
```

- `git reset --hard <commit>` reverts code back to a previous commit
- `git reset --hard origin/master` reverts code back to remote repository version

Branching

- allows you, in a single repository, to have a couple differenct kind of versions of the code that are going on simultaneously
- branch is a version of the repository
- each branch has its own commit history and current version

```
git branch
```

- show all branches of code
- create a branch with `git branch <branch_name>`
- switch to (*checkout*) a new branch with `git checkout <branch_name>`

```
git merge
```

- used to merge two different branches together
- `git merge <branch_name>` merges the branch *branch_name* with current branch

Pull Request

- let you tell others about changes you've pushed to a branch in a repository on GitHub.
- Once a pull request is opened, you can discuss and review the potential changes with collaborators and add follow-up commits before your changes are merged into the base branch.