Remote Git Repo - #7 Collaboration

#git-workshop

Cloning

Cloning is one way of downloading a repo from GitHub (or any remote URL).

```
$ git clone https://...
```

This is the same as doing a `git init`, `git remote add`, and `git pull`.

Multiple Local Repos

We'll have multiple local versions of the repo floating around now. One consequence of this is that they won't necessarily all be the same. Let's say you and a friend have cloned a copy of this repo.

Now, the problem arises when you both make changes on the master branch. Let's say you make the following changes:

```
# Person 1
$ echo "Remove all pangrams!" > pangram1.txt
$ git add pangram1.txt
$ git commit
```

And your friend does the following:

```
# Person 2
$ echo "TODO: add a pangram here" >> pangram1.txt
$ git add pangram1.txt
$ git commit
$ git push origin master
```

Now when you try and push to GitHub, you will be met with an error message. In this case, your changes conflict with the commit your friend just pushed.

Resolving Merge Conflicts

This is very much the same as resolving local merge conflicts, with the one big difference being that you'll probably want to push your merge commit when you're done.

We'll open the conflicting file in a text editor and clean up. Then, do the same stuff as before:

```
$ git add pangram1.txt
$ git commit
$ git push origin master
```

This time, your changes should take.

> In general, it's a good habit to push and pull frequently to limit the *time lag*, which ensures your repo and the remote repo do not become too outdated.