#4 GitHub Collab

#advanced-git

Fetch

In the last workshop, we covered `git push` and `git pull` as tools that would keep you in sync with a remote repo. In truth, the `git pull` command usually combines a `git fetch` and a `git merge`. This is an unimportant detail for repositories with simple histories.

In reality, there are local branches in your Git repo for every remote branch. For example, if you've pushed to the `master` branch on the remote named origin, you'll have a `remotes/origin/master` *remote tracking branch* (`origin/master` also works). The following command shows all branches, including remote tracking ones:

```
$ git branch -a
* master
  quack
  remotes/origin/master
```

By default, `git fetch` will update all the remote tracking branches for `origin`. If you have other remotes, you can fetch them by supplying the name:

```
$ git fetch remote_name
```

Or to fetch all remotes;

```
$ git fetch --all
```

Standard Workflow

Of course, standard workflow will be a little different for everyone. But here are some things that are fairly common practice.

Work on your own branch

There are many reasons to keep your work to yourself, especially if it's part of a larger project. Branches are also fairly easy to create in Git, though merging can be tricky.

Keep up to date

If there are updates to the master branch, you'll probably want to take them into account when working on your feature. There are ways to do this with a merge and with a rebase. Whichever you prefer is up to your team and your personal preferences (yes, in that order).

Merge:

```
$ git checkout master
$ git pull origin master
$ git checkout feature
$ git merge master
# Shortcut
$ git checkout feature
$ git pull origin master
```

Rebase

```
$ git checkout master
$ git pull origin master
$ git checkout feature
$ git rebase master
# Shortcut
$ git checkout feature
$ git pull origin master --rebase
```

Push your work at the end of the day

Remember that your commits are local only, until you push them. If you want others to see your work, make sure you push regularly! In fact, you might even want to push after every commit (though this may be a bit extreme).

Pull Requests

Pull requests (sometimes known as merge requests) are a more formal way to merge branches, usually into master.

Tags and Releases

By default, `git push origin master` will not push tags. To do so, you must supply the tag name:

```
$ git push origin tag_name
```

To push all tags:

```
$ git push --tags
```

> We didn't learn about the difference between annotated and 'lightweight' tags, but it may be interesting to read up on: Push a tag to a remote repository using Git? - Stack Overflow In general, annotated tags are better for when you create releases since they have important metadata such as author names and messages associated with it.

Pushed tags will show up on GitHub under "releases". This is a popular way for projects to show end users what version of the software they should download.

Changing History and Remotes

Git history can be broadly partitioned into two parts. *Public history* is everything that has been pushed to a remote repository. *Private history* is commit history that exists only locally. A brief comparison:

Public history is

- Backed up and less likely to be lost (although not impossible!)
- More troublesome to change

Private history is

- Local only and usually hidden from other collaborators
- Easier to play around with, and carries fewer consequences if changed

It's a good habit to avoid changing public history if possible. If you must, try to limit to public history that only you work on (for example, a branch that you own). To update public history after changing it, it's necessary to use a force push.

Force Pushes

When a branch is rebased, the history of that branch changes. It will become necessary to

force push the update to the remote. If any others are using your branch, they will likely become very upset (another good reason to work on your own branch)! This can be avoided by just merging when pulling updates from master.