All About Branches

Branch name inconsistency

Sometimes the branch name shown in `git` are inconsistent.

For example, the output for `git branch -a` vs `git branch -r`. The remote-tracking branches will be prefixed with `remotes/` in `git branch -a`. However, the remote branch in `git branch -r` won't.

Remote

A remote you can just think of as a repository location that is somewhere else on another machine. You refer to remotes using a URL and that's where the remote repository is stored.

You want to store your repository on another machine for backup and team collaboration purposes.

Remote name

Names like `origin` that you set up for your remote tracking repository just refers to a URL that you remote repository lives under. If you are going to be referring to it with URL every time it will be super time consuming. Think of it just as alias to URL.

Branches on your machine

You got three types of branches on your local machine **non-tracking local branches, tracking local branches, and remote-tracking branches**.

On the remote machine (remote repository), you only got one type of branch.

1. Local branches

Running `git branch` you will get a list of all local branches on your machines.

`.git/refs/heads/` file also contain list of local branches.

You got two types of local branches.

1.1 Non-tracking local branches

These branches are not associated with any remote repositories, not "synced" per se.

You would create one by running `git branch <branchname>` or `git checkout -b <branchname>` to create and switch to the branch.

These branches are also the one that you need to set up upstream for, i.e. if you do `git push` it will complain about not having a upstream and you would have to run `git push -u <remote> <branch>` in order to set it up.

1.2 Tracking local branches

Tracking local branches are associated with another branch, usually a remote-tracking branch.

You can check which one of your local branches are tracking branches using `git branch -vv`:

```
git branch -vv
  feature/b/lol a54cf89 [origin/feature/b/lol] Create player
* kms           a54cf89 Create player
```

You can see that my local `feature/b/lol` branch is tracking `origin/feature/b/lol`.

Tracking local branches are useful in the sense that they allow you to run `git pull` and `git push` without having you to specify the remote and the upstream branch to use. If you don't have the tracking branch set up and you try to `git push`, Git will complain about not having an upstream branch, you would just set it using `-u <remote> <branchname>` and then it will set the tracking branch for you automatically.

2. Remote-tracking branches (still on your local machine)

To see the list of remote-tracking branches on your machine by running `git branch -r`

`.git/refs/remotes/<remote>/` contains the list of remote-tracking branches

They are a "local" copy of branch that the remote repository contains. You would be updating these branches by using `git fetch` or via `git pull`. This is what your tracking local branches will be using to compare and see whether you are behind or ahead of your remote repository.

Although remote-tracking branch is stored locally on your machine, it isn't really referred to as a local branch.

Git branch cheat sheet

- To delete a local branch (regardless of tracking or non-tracking):
  `git branch -d <branchname>`
- To delete a remote-tracking branch (not done often):
  `git branch -rd <remote>/<branchname>`
- To create a new local non-tracking branch:
  `git branch <branchname>`

Revision #2
Created 2023-03-10 17:20:46 UTC by Tamarine
Updated 2023-03-10 19:50:38 UTC by Tamarine