Lesson 3 - Using GitHub

Jithin K. Sreedharan

- GitHub is a website that makes it easy to share an entire git repository with other people.
- The local git repository needs to be synced to the GitHub repository. GitHub uses the concept of remote repository (simply *remotes*), that's the repository in Github. One only needs to commit a branch (called push branch). All the commits under that branch (by tracing parents) are synced to the GitHub repository.
- Looks like Github clone (`git clone`) is not possible from local repository to the Github server.
- Adding a remote:
  - Go to https://github.com/new. Create a new repository.
  - Add the GitHub repository as a remote to the local repository (here "reflections") in my computer

```
git remote # View current remotes
git remote add <name> <URL>
#"Name" is any name that one can use to refer to the remote repository.
# If there is only one remote, standard name is "origin"
# Copy the HTTPS URL, not SSH URL from the GitHub info.
git remote add origin https://github.com/jithin-k-sreedharan/reflections.git
git remote # To see the added remote
git remote -v # To see the details of added remote, "v" for verbose
git push origin master # git push <origin> <master>
```

- For making the GitHub repository not to ask for password each time we push, follow this link.
- Reflections: When would you want to use a remote repository rather than keeping all your work local?
  - To keep safe all my code and data of a project in a cloud
  - To share and collaborate projects with other people
- Pulling from the remote to local repository

```
git pull origin master # git pull <remote_name> <branch_name>
```

- Reflections: Why might you want to always pull changes manually rather than having Git automatically stay up-to-date with your remote repository?
  Having some control over what to pull and not unexpectedly change the files in the local repository.

- Forking:
  - It is equivalent to cloning an existing GitHub repository inside GitHub (may be into another GitHub account)
  - Cloning: copying an existing repository either from remote repository or from local repository.
  - Making any changes in the forked repository will not affect the original repository.
- When you clone a remote repository to your local repository, git automatically sets up a remote pointing to where you clone from.

- Reflections: Describe the differences between forks, clones, and branches. When would you use one instead of another?
  - Forks: Clone from GitHub repository to another GitHub account
  - Clones: Copy from a repository (remote or local) to another repository (probably local)
  - Branch: Inside a repository, make a diversion from the existing code base
- **Merging and resolving conflicts between remote and local repositories**:
  - The local repository has a hidden branch called `origin/master` (name_of_the_remote/remote_branch_name).
  - Look at the scenario when the local repository and remote repository made changes and they have not synced yet.
  - Now when the local issue the command `git fetch` the origin/master will get updated the latest change of the remote. Then to merge origin/master to master of the local repository, one needs to issue the command `git merge master origin/master` (merge should be issued after moving to master branch and master as the first argument of git merge command)
  - `git pull origin master` carries out both the commands `git fetch` and `git merge master origin/master` together.
  - `git merge master origin/master` might issue a CONFLICT merge fail message. Make the necessary changes in the associated file. Then issue `git add <conflict_file>` and `git commit`. After all of the above, type `git push origin master` to push the changes to the remote. Finally `git status` should show up-to-date.
- Reflections: What is the benefit of having a copy of the last known state of the remote stored locally?
  It helps not to lose the changes made by the remote. Having origin/master makes the system to pull the new changes only to this branch.
- `pull-request` in GitHub is actually `merge-request` to request the administrator to merge a new branch to the master or any other old branch. `pull-request` can be even made to the master branch of the original repository from which we forked the contents.
- Reflections: How would you collaborate without using Git or GitHub? What would be easier, and what would be harder?
  I might use Dropbox, but tracking changes and versions might be difficult and requires lot of manual efforts.
- Suppose I make a fork of a `repository1`. Later I created a new branch called `change` in my local and remote repositories. Now I want to merge `change` branch into `repository1`. But it's not possible to initiate a pull-request, as the original `repository1` has changed a lot since I forked from it. Resolve the issue with the following steps:

```
git remote add upstream <URL_of_repository1>
git checkout master
git pull upstream master
git checkout change
git merge master change
# Merge conflict in changed_file
git add <changed_file>
git commit
git push origin change
```

- GitHub documentation on syncing a forked repository with the original one: https://help.github.com/articles/syncing-a-fork/.
- Documentation of `.gitignore` file
  https://labs.consol.de/development/git/2017/02/22/gitignore.html
  https://git-scm.com/docs/gitignore
- Lessons learned:

- Used remotes to push changes up to Github and pull down changes made by other people.
- Practised pull-request to collaborate with other people.