Exercise 2: Your First Remote

Overview

In this exercise you will connect your local Git repo to a remote repository on GitHub.
You will learn two ways of initiating this connection: 1) Create a remote copy of your local repo and 2) copying a remote repository onto your local disk. You will push your example file to both your own remote and our collaborative remote. These are the important commands to move files between local and remote:

```
git remote -v
git remote add <remote_name> <url>
git clone <url>
git push <remote_name> <branch_name>
git pull <remote_name <branch_name>>
```

1. Pushing an existing local repo to GitHub

Go to GitHub and create a new public repository there. You don't need to initialise with a README since we already have created some file locally.

example_newrepo

On clicking the "Create repository" button, the GitHub will ask if you want to create a new repo from scratch or if you want to push an existing repo you have created locally. Since we have already created a new repo locally, we want to push that onto GitHub. So all we need is to copy the URL.

In order to push the local repository to our GitHub repository, we need to set the remote URL of the newly created GitHub repository to our local repository. The command "git remote -v" will show whether our local repo is linked to any remote URL.

```
>> git remote -v
>>
```

No output means, by default the remote URL is not setup for this local repository. So, we will setup the remote URL using:

```
>> git remote add origin https://github.com/your-username/your-repon.git
```

Double check whether this worked by trying `git remote -v` again!

```
>> git remote -v
origin    https://github.com/aranas/my_first_remote.git (fetch)
origin    https://github.com/aranas/my_first_remote.git (push)
```

Now, we can try pushing the contents of our local repo to the remote GitHub repository. It is standard to add the main remote repository under the name "origin" and the default branch is always called the "main" branch. So the following command translates to copying all contents to the main branch of the remote called origin. The `-u` flag will automatically set up the remote main branch to track your local main branch.

```
>> git push -u origin main
Counting objects: 3, done.
Writing objects: 100% (3/3), 263 bytes | 263.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
```

```
To https://github.com/aranas/my_first_remote.git
 * [new branch]      main -> main
(base) Sophies-MacBook-Pro-2019:test sophiearana$
```

Now, since all your files are on GitHub, you can edit them directly online as well. Refresh your browser, open your name_surname.md file and edit it by clicking on the pen-icon

example_editgithub
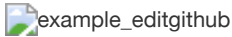
As you will see GitHub provides you with a GUI for editing and commiting your changes to the remote repo. Now, make a change and click on the commit button!

Next you will have to get those changes into your local folder. For this you will use the `git pull` command. This command is actually a shortcut for running `git fetch` followed by `git merge`. But you don't need to worry about this right now.

```
>> git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/aranas/my_first_remote
 * branch            main      -> FETCH_HEAD
   9770d9c..2ddf88b  main      -> origin/main
Updating 9770d9c..2ddf88b
Fast-forward
 John_Doe.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Congratulations! You have now successfully synced your local and your remote repository. Think of it as a collaboration with yourself :)

2. Cloning a remote GitHub repo to your local disc

Sometimes you won't have a local project to start from but rather, you would like to copy an already existing remote project to your own disk. In that case, you want to `git clone` the remote repository. For our next exercise we will jointly work on our workshop repository https://github.com/Donders-Institute/git-workshop.git in small teams of five. To get set-up, your first step is to clone our remote repository

```
>> cd ~/git_workshop2020
>> git clone https://github.com/Donders-Institute/git-workshop.git
Cloning into 'git-workshop'...
remote: Enumerating objects: 157, done.
remote: Counting objects: 100% (157/157), done.
remote: Compressing objects: 100% (88/88), done.
remote: Total 157 (delta 48), reused 152 (delta 46), pack-reused 0
Receiving objects: 100% (157/157), 16.15 KiB | 3.23 MiB/s, done.
Resolving deltas: 100% (48/48), done.
```

You can see that a new folder git-workshop was created containing the repository and all remote connections are automatically set

```
>> cd git-workshop
>> git remote -v
origin    https://github.com/Donders-Institute/git-workshop.git (fetch)
origin    https://github.com/Donders-Institute/git-workshop.git (push)
```