

Technology Research Preparation Pre-work 2

Collaboration using Github

Lu Qin

Autumn 2016

1 Pre-work 2 for Assignments 2 and 4

Both Assignment 2 and Assignment 4 are teamworks for the professional stream, in which you need to collaborate with other team members to finish a literature review (Assignment 2) and a project proposal (Assignment 4). Your third task is to learn a wonderful tool for teamwork - Github. Since your team has not been established, you are only required to do the jobs as a **team leader**.

1.1 Before using Github

Github is a powerful tool for version control and teamwork. Before you use Github, please go to the Github website¹ and register an account. You also need to download and install Git (the command line tool for Github) from the Git website². After installing Git, you can open a command line terminal and run the following commands:

```
git config --global user.name your_user_name_in_github
git config --global user.email your_email_in_github
```

A good tutorial for Github and Git can be found in Udacity³. Below, let's see some simple ways to use Github and Git for a teamwork.

1.2 Using Github and Git for Teamwork

In this team, there are three team members who need to collaborate to write a literature review. Suppose the Github account names for the three members are Jim, Kate, and Lucy, while Jim is the team leader. In the following, let's see how Jim, Kate, and Lucy can collaborate as a team to finish the literature review. Let's suppose each member in the team has a local computer, and the team uses Latex to complete the literature review. The teamwork can be finished in three phases.

¹<https://github.com/>

²<https://git-scm.com/downloads>

³<https://www.udacity.com/course/how-to-use-git-and-github--ud775>

- Phase 1: Team leader Jim builds the framework.

In the first phase, Jim, the team leader, may do some work to make sure the cooperation can be done successfully. He can do this in three steps.

- Step 1: Login Github⁴ and create a new public repository and name it “JKL”. The link of the repository should be like the following:

`https://github.com/Jim/JKL.git`

Then, in the JKL repository in Github, go to “settings” → “collaborators” and add two collaborator one by one by entering each user name of your team members. For this team, Jim need to add two collaborators “Kate” and “Lucy”. (Since your team has not been established, you can do this afterwards).

- Step 2: Go to your local computer, create a folder named “JKL”, and create six empty files, named “Jim.tex”, “Jim.bib”, “Kate.tex”, “Kate.bib”, “Lucy.tex”, and “Lucy.bib”, and then create a file “Final.tex”, which will be the final version by integrating the works of all team members together. You can download a sample “Final.tex” and other six files in “JKL.zip” from UTSONline. Here, the three team members have the following responsibilities:

- Jim is responsible to edit “Jim.tex”, “Jim.bib”, and “Final.txt”;
- Kate is responsible to edit “Kate.tex” and “Kate.bib”;
- Lucy is responsible to edit “Lucy.txt” and “Lucy.bib”.

Don’t worry, you will see how to do this later in Phase 2.

- Step 3: Open a command line terminal and go to the “JKL” folder you have just created, and then run the following commands:

```
git init
git add *
git commit -m "initial commit"
git remote add origin https://github.com/Jim/JKL.git
git push origin master
```

The above commands allow you to upload all the files in your local “JKL” folder to the “JKL” repository in the Github server. You may need to be asked to enter your account name and password to upload your files to the server.

- Phase 2: Each team member write their own part individually.

Now Jim has done something to build the framework for collaboration. Other team members Kate and Lucy also need to create the “JKL” folder in their local computers, open the command line terminal, and then run the following commands in the “JKL” folder.

⁴<https://github.com/>

```
git init
git remote add origin https://github.com/Jim/JKL.git
git pull origin master
```

The above commands allow Kate and Lucy to download the files Jim created into the “JKL” folder in their local computers.

After doing this, Jim, Kate and Lucy can edit their files in their local computers individually. Each time they do some changes on the files that they are responsible for, they can follow three steps to make their changes updated on the server.

- Step 1: Open a command line terminal and go to the “JKL” folder in your local computer, and then run the following command to download the latest version of all files.

```
git pull origin master
```

- Step 2: Edit the files that you are responsible for. Do not edit the files for others. Only the team leader Jim can edit “Final.tex” and compile the file to pdf.
- Step 3: In the “JKL” folder, run the following commands to upload your local files to the Github server and make your local files up-to-date.

```
git commit -a -m "adding a new reference"
git pull origin master
git push origin master
```

Here the message “adding a new reference” can be replaced by any message to describe the new change.

- Phase 3: Team leader Jim integrates the works of all members.

Since this team is using Latex, and the Final.tex has already integrated the works of all members together. The team leader Jim can compile Final.tex to pdf using the following commands:

```
pdflatex Final
bibtex Final
pdflatex Final
pdflatex Final
```

However, if you are using word, it is not easy to automatically integrate the works of all team members. In this situation, the team leader can manually create a file such as “Final.docx” to integrate the works of all team members together.

1.3 Using other commands of Git

You can use the following command to see a entire list of commit history of the “JKL” repository.

```
git log
```

Of course, this is just a simple example of collaboration using Github and Git. You are more than welcome to use more advanced commands of Git for version control and collaboration in a teamwork.