SE 206 Git Lab 2016

By SESA

Step Zero - GitHub Account

- Go to https://github.com
- Create an account and associate your uni email with it
- Sign up for the Github Student Pack

Step One - Git Setup

- Set up Git

```
$ git config --global user.name "John Doe"
```

```
$ git config --global user.email jdoe123@aucklanduni.ac.nz
```

```
$ git config --global core.editor vim
```
(or your favourite editor)

- Create an ssh key

```
$ ssh-keygen -t rsa -b 4096 -C "jdoe123@aucklanduni.ac.nz"
```

You will be prompted to enter a filename (press enter to use the default) and a passphrase (you should use a strong password you will remember).

- Print the public key to your terminal and copy it

```
$ cat ~/.ssh/id_rsa.pub
```

- Add the key to your github account (https://github.com/settings/keys)

Adding this key to your github account will allow you to prove your identity without typing your github username and password every time you want to perform an action.

Step Two - Create a Local Repo

- Create a new directory and initalise an empty git repo inside it.

```
$ git init
```

- Write a hello world program in your favourite language (you must be able to run it easily on uni linux without root access)
- Create a .gitignore file for this language/platform (https://github.com/github/gitignore)
- Add your hello world program to your tracked files

```
$ git add .
```

- Commit your changes

```
$ git commit -m "Enter a useful message here"
```

- Create a file called README.md with instructions on how to build and run your project
- Add and commit your changes

Step Two - Get Your Project Onto Github

- Create a repo on github
- Add your github repo as a remote (the url you need should be on the page that shows up once you have created your repo)

```
$ git remote add origin enter your ssh clone url here
```

- Push your changes to your remote repo

```
$ git push -u origin master
```

Step Three - Collaborate With a Friend

- Find a friend/person sitting next to you
- Ask them to send you a link to your project on github, give them a link to yours
- Press the fork button on their project page on github to make a fork of it in your own account
- Clone your fork

```
$ git clone git@github.com:username/projectname.git
```

- Try to follow your partner's instructions to build and run their project
    - If you can run it easily, great
    - If you can't run it easily, somebody isn't very good at following instructions
- Now add your own hello message to their project and add and commit your changes
- Push your changes to your fork of the repo

```
$ git push
```

- Create a pull request back to the original repo by going to the pull requests section of the repo page on Github

Pull requests can be also made between branches if you are both working on the same repo in Github, forks are for when the owner of the project does not want to give you access to the original repo

- Get your partner to review and accept your pull request (if it's good enough)
- Meanwhile, review and accept their pull request

If you are working with more than one other person, you may get merge conflicts at this stage, if this happens, it is the PR submitter's responsibility to fix said conflicts. To do this, pull the source repo's master branch into the source repo's master branch and

fix the conflicts that arise, then run `git add` and `git commit` and push back to your fork repo, this will add a commit fixing the merge conflicts to the pull request.

What Have We Done?

To have completed this lab, you should have the following:

- A repo on your local machine that prints hello world in your favourite language.
- A repo on your github account containing this project, it should have two commits by you and at least one by a friend.
- A fork of someone else's hello world repo.
- A commit on attributed to you that has been merged into the original repo