Practical 4. Getting started with Git and GitHub

Rob Young

IBI1, 2020/21

1 Learning objectives

- Explain the principles of version control
- Manage projects with git and github
- Build a simple website using github pages

2 Git

- Here, you will start a small version control project and learn to work with Git
- Log into the ZJE server as in Practical 3. Create a directory IBI1_2020-21 within your home directory and move into that directory.
- Start a local project by typing `git init`. You should see the following message on your terminal window: "Initialized empty Git repository in /public/workspace/<USERNAME>/IBI1_2020-21/.git/".
- Create a new file called 'README.md' using the text editor nano by typing `nano README.md`. Edit this file to include a few lines describing the project. For instance: "This is my first git project." Once you have added the text, exit nano and save the file. You do this by (1) typing `Ctrl-X` or `Cmd-X` in macOS (2) tying Y to choose yes (3) pressing Return to confirm your choice.

  (There are lots of other options you can use in nano which you might like to explore outside of the practical session. This weblink contains a lot of information that you might find useful in becoming more efficient in using nano: https://www.howtogeek.com/howto/42980/the-beginners-guide-to-nano-the-linux-command-line-text-editor/)
- Add this file to your repo by typing `git add README.md`. You have added a file to your repository but you haven't yet committed the changes. Commit the changes with the command `git commit -m "Initialise"`. The -m option adds the commit message so you can set it to a different message if you like. You should see an output message stating that you have 1 file changed with 1 insertion.
- Look in your log file by typing `git log --oneline`. Can you see your initial commit?
- Now it's time to create a new file. Use your text editor or some other method to create a new text file, **thingsIlike.txt**. Use this file to make a list of 10 things you like, in this format:

  I like tea
  I like chocolate biscuits
  I like snow…
- Add this file to your repository and commit these changes. What does your log look like now?

- Now, change your mind! Add one extra thing you like. Commit by typing `git commit -m "edit thingsIlike"`. Does this error message make sense? Now add the modified file and try committing again.

- Now, change your mind again! Commit back to the previous version (the original list of 10 things). You can do this by typing `git revert HEAD`. This command displays a commit message which you can escape by typing `:x` (full disclosure: this message is displayed in another text editor vim, which is similar to nano). What happens next? How does your log change?

- Create a new branch with a friend's name (e.g. Alex) by typing `git branch Alex` and switch to the friends branch by typing `git checkout Alex.`
- In the friend branch, remove 3 things you don't like. Add 5 things you like. Change one "like" to "don't like". Add and commit these changes as before.
- Move back to the master branch (`git checkout master`). Make a few changes to your favourite things yourself: Remove one thing, add one thing, change one "like" to "like very much"
- Merge the friend branch into your master branch. First move back to the master branch if you have not already done so and then type `git merge Alex` (assuming that the new branch you made was called Alex). You may need to escape the commit message as previously for `git revert`". What does thingsIlike.txt look like now?

3 Getting started with GitHub

- Here, you will start a small version control project and learn to work with Git
- Go to GitHub: https://github.com/
- Sign up. You will be asked to choose a user name, and to provide an e-mail address and a password. Your username can be anything (as long as it is not already taken), but bear in mind that you will probably have your GitHub account for a long time, so choose a name that is sensible. (Bear in mind also that your instructors will see your GitHub repo!).
- You will be asked to solve a "Captcha" puzzle to prove that you are a human, not a bot. Once you have done this, choose your plan. We will be using the "Free" version in all of our courses, so please select this. Click "Continue". There is a little questionnaire (which you can skip). You will then be asked to verify your e-mail address. Check your e-mail (including your spam folder) for a message from GitHub, and follow the instructions. Congratulations, you now have a GitHub account.
- At the moment, you don't have any repository in GitHub. In this practical, you will make two: One by forking an existing repository, and one by creating your own (see below). We will start with the fork. In the search box in the upper left, search for IBI1_2020-21. You will be shown the **IBI1_2020-21**. Click on it. You can have a look at the file structure etc. Click "Fork" (top right) to create a fork of the repository. This may take a bit of time. But when it's done you will have your own repository called **IBI1_2020-21**
- You have successfully created a repository in GitHub, but to really be able to work with it, you want a copy of it on your own computer. You can do this by typing the following command: `git clone https://github.com/r0bah0lic/IBI1_2020-21`. You will be asked for your GitHub

account username and password. You may have to wait a few seconds for everything to download but afterwards you should see an `IBI1_2020-21` directory. Is the file structure the same as you saw on GitHub?

- Move into this directory and have a look at the repo. How many commits has it had so far? What other information about previous commits can you see?
- Now is your time to make this directory your own. Find the file **Practical3/Introductions.txt**. Edit it and replace the information about Rob with information about yourself. Commit your edits.
- Copy the file **thingsIlike.txt** you made in Part 1 into this repository (into the directory called **Practical3**). Commit your changes. (This is not something you would normally do very often, but in this case it makes sense).
- Now, how to move your local changes back to GitHub? Type the following command `git push origin master`.
- Go back to your browser and look at your GitHub again. You should see that the content of your repository has changed. In directory **Practical3**, you should see that the file **Introductions.txt** has recently changed, and you should see your own commit message. You should also see the additional file **thingsIlike.txt**. Compare this to the original repo you forked from: https://github.com/r0bah0lic/IBI1_2020-21. This has not changed. (That's the way it should be. Your fork is for you to edit and play around with, but it will not change the "Master" copy of practical instructions).

4 Making a website with GitHub pages

GitHub can also be used to make a simple website. We will do this here.

- In GitHub, create a new repository by clicking on the "+" symbol in the upper right corner, and then selecting "New repository". Your repository needs to have a very specific name: ***username*.github.io**, where ***username*** is your github user name. When you create your repository, you can add a description if you want. Make the repository public. Do not click "Initialize this repository with a README".
- Now it is time to choose a style for your website. Here, we will use a pre-defined style through a framework called Jekyll. In your ***username*.github.io**, go to "Settings". It's OK to leave the options as they are. Go to the "GitHub Pages" section and click on "Choose a Theme". You will see a gallery of possible designs on the top. Click on one to see what it would look like on a website. Pick one you like and click on "Select theme"
- Now if you go back to the "Code" tab on your GitHub repository, you will see that a new file has appeared, called **_config.yml** - this stores the information necessary to achieve the design you are looking for.
- Move back to your home directory and clone your ***username*.github.io** repo.
- There is a file called **index.md** in **IBI1_2020-21/Practical3**. Copy that to your ***username*.github.io** repo. Replace the old file. Commit. Push the changes to GitHub. In your web browser, type ***username*.github.io**. You can also find the link in "GitHub Pages" section. What do you see? Look at the **index.md** file in your text editor or in GitHub. Do you understand how it works? Can you edit it, so your web page shows information about yourself? Here is an example we did: https://r0bah0lic.github.io

5 For your portfolio

The markers will look for and assess the following:

- A simple website about yourself at ***username*.github.io** with some kind of styling (no plain text), and a little bit of text that is different from the index.md file you were given. You can use this as your personal webpage and add whatever information you like. It does not need to stay the same as it is at the end of this practical.

- In your fork of **IBI1 _2020-21/Practical3**, you should have a file called **Introductions.txt** that has been edited by you and that contains information about yourself.

- In your fork of **IBI1 _2020-21/Practical3**, you should have another file called **thingsIlike.txt** that contains a list of things you like.

- In your fork of **IBI1 _2020-21/Practical3**, please also add a file called **Reflection.txt**. In this file, briefly note your thoughts about the Practical. Did everything go well? Are there things that were challenging? What questions do you have? We will discuss your questions in tutorial 3.

- The commit messages on your fork of **IBI1_2020-21/Practical3** should be relatively informative. (They don't have to be perfect, but they shouldn't all be empty, and they should have information about the kind of change that was made, not just "Commit 4" or "blablabla")

You can add or edit things after the Practical session. We do not look at the commit date, we just want it all to be there!