Practical 3. Getting started with Git and GitHub

MI Stefan

IBI1, 2018/19

1 Learning objectives

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

2 Git and GitKraken

- Here, you will start a small version control project and learn to work with Git and GitKraken
- Open GitKraken. Go to **Start a local project**. You will have to choose a name for your repository, and also a directory where it will live as a sub-directory. (You can select that in the **Initialize in** field). Once you have created your repository, you should see that you have exactly one branch (master) and exactly one commit (initial commit).
- You will also see that a file has been created called **README.md**. README ("read-me") files are useful for short descriptions of projects. If you encounter an unfamiliar project, you can check the README file for useful information. In this case, GitKraken has helpfully created a README file for you. It is empty, except for the first line, which contains the name of the project.
- Edit the README file. You can do this from within GitKraken (using **Edit this file**) or outside of GitKraken in a text editor (e.g. Notepad). Add a few lines describing the project. For instance: "This is my first git project." Once you have added the text, save the file (according to how this is done in your text editor).
- Look at your repo in GitKraken. You will see that something has changed. There is a new circle with a dotted line around it in your log, and it has the letters "WIP" (Work In Progress) next to them. This indicates that something has changed, but you haven't yet committed the changes. Commit. In the right-hand menu, click on "Stage all changes". In the "Commit Message" field, write a brief commit message (E.g. "Added description to README"). Click "Commit changes". What has changed in your log?
- 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 cheese
  I like music
  I like sunshine
  ...
  ```

  Commit the changes you made.
- Now, change your mind! Add one extra thing you like. Commit.

- Now, change your mind again! Commit back to the previous version (the original list of 10 things). You can do this in GitKraken by navigating to the commit you are unhappy about, bringing up the context menu using your right mouse button and clicking "revert commit". What happens next? How does your log change?
- Find a friend among the students sitting around you. Create a new branch with your friend's name. They should do the same thing on their computer. Move to the "friend branch". Exchange computers.
- Friend, remove 3 things you don't like from your friend's list. Add 5 things you like. Change one "like" to "don't like". Commit your changes. Hand the computer back to its owner.
- Owner, move back to the master branch (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 your friend's branch into your master branch. You may be told there is a "Merge conflict". Click on the filename. This shows you the differences in both branches side-by-side and allows you to pick the bits from each branch that you want to keep in the final output file. You can select entire sections by clicking in the box next to them, or single lines of text. Create a version you are happy with and merge (and commit!)

3 Getting started with GitHub

- 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 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 2018-19. You will be shown the **IBI1_2018-19** repo that Mela has created. 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_2018-19**.
- 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 through GitKraken. First of all, you need to let GitKraken know about your shiny new GitHub account! Go to **File** → **Preferences** → **Authentication** → **GitHub.com**. Click "Connect to GitHub" and follow the steps.
- Now, in GitKraken, go to **File** → **Clone Repo** → **GitHub.com**. You will have to choose a local directory (on your computer) that will house the local repo. You can also choose the "Repository to clone" from the list of your GitHub repos. Click "Clone the repo".
- 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 Mela with information about yourself. You can do this directly within GitKraken, or using a text editor of your choice. 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? Simple! Click "Push".
- 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/MelanieIStefan/IBI1_2018-19` - 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.
- Use Gitkraken to clone the repo.
- There is a file called **Readme.md** in **IBI1_2018-19/Practical3**. Copy that to your **<username>.github.io** repo. Commit. Push the changes to GitHub. In your web browser, type **<username>.github.io**. What do you see? Look at the **Readme.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://melanieistefan.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 Readme.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_2018-19/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_2018-19/Practical3**, you should have another file called **thingsI-like.txt** that contains a list of things you like.
- In your fork of **IBI1_2018-19/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_2018-19/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!