**Learning Git lesson plan**

Git allows you to take snapshots of your work to save all you did. If in the future you need to go back to a stable version of your program, you’ll be able to easily revert. It also allows you to easily collaborate with other people while working on projects together.

If they don’t already have a Github account, go to Github.com and make an account

Open up GitKraken. Should be in the Data drive

GitKraken is a git GUI (Graphical user interface). We use this as it’s easier than using the command line which is blocked on the school computers

Initialize a repository

1. We need to create a folder where the repository will be kept.
2. Open notepad++
3. Save the blank file as index.html in a new folder
4. Go to GitKraken and click the folder in the top left
5. Init>Local Only>Browse>Find your folder
6. Click Create repository. Don’t add the license if you want. DON’T USE GIT IGNORE
7. Now that it’s initialized, click where it says //WIP. Find the button that says stage all changes, write the commit message and click commit
   1. Stage all changes allows you to select the changes you want to have committed to this current commit. It gives you lots of flexibility so you don’t commit something you don’t want for any reason.
8. Now we can go back a commit. Right click Initial Commit and select Revert commit. Now all the things you did are gone so you are back at a “stable” version

Commits is the most used thing in git. All it is, is writing what changes you made and selecting the files that you changed you want to put in that commit. All your commits are shown in the middle of the screen

Branches

1. When you want to add a new feature, it’s best practice to use a branch
2. Create a branch by clicking branch at the top
3. Now it asks you to name your new branch in the commit section. Name add-feature the quick brown fox jumps over the lazy dog the quick brown fox jumps over the lazy dog the
4. Now you’ve switched to this new branch, you can do anything you want without damaging the master branch (the stable branch)
5. Let’s go to index.html and make whatever changes you want. Then commit those changes
6. Change back to the old branch by double clicking master on the side tab in GitKraken
7. Notepad++ should say a program changed the file, you should click yes.
8. Now you should see all your changes are gone. It does this as you are back in the master branch.
9. To have all your changes put back, double click the branch add-feature.
10. Now we’ll merge this branch back into master as we added what we wanted
11. Under branches, click the 3 dots beside master and click merge add-feature into master
12. Now you see in the commit section, it looks different. There is a branch from the original part that goes back into the original part.
13. Since we’re done with the add-feature branch, we can delete it.
14. Switch back to the master branch by double clicking it
15. Click the 3 dots beside add-feature, then click delete add-feature, then confirm deleting it at the top of the screen

Remote repositories

How do we share our work with others and how are we able to work with others like this?

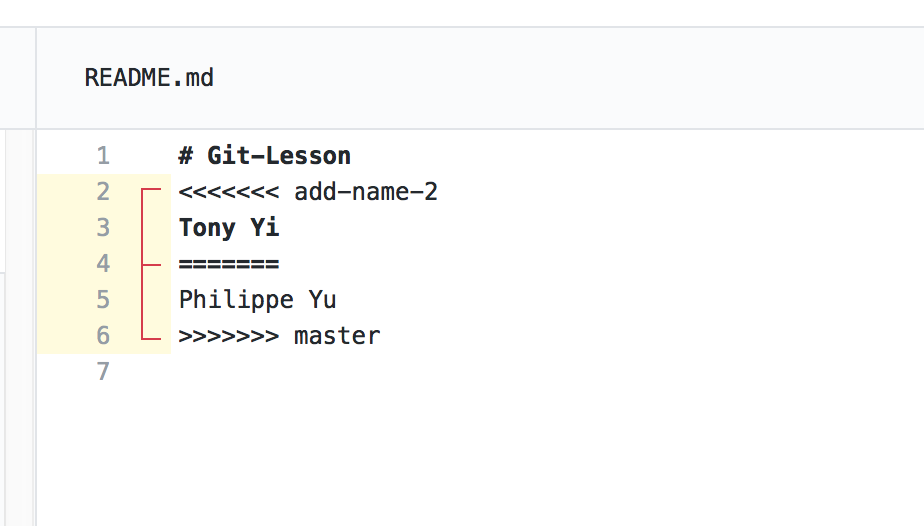
In order to do that, we need to upload this to a remote repository on the internet.

Most people use Github for this as it is very popular and is where almost the entire open source community is.

1. Go to Github.com and login
2. Click new repository and name it git-lesson, or whatever you named the file.
3. Write a description if you want and DO NOT check initialize this repository.
4. Now it gives you a quick setup page. Copy the https link and open GitKraken
5. Under remote, click the big + button
6. Now it’ll ask you for the remote. Name it origin. Make the push and pull url the one you just copied.
7. Now we have a new remote here. All you need to do to trigger the upload is click push
8. It’ll ask you “What Remote/Branch should “master” push to and pull from? Don’t change anything and click the green submit button
9. It’ll now upload to Github and if you refresh the Github.com page, you’ll see all your things are there.
10. Click the big Add a README button and then scroll down to where it says Commit new file.
11. Write Added README.md then click commit new file
12. We have these changes in the remote, but not on your computer. How do we download it from the server?
13. Go to GitKraken and click pull. This will download it and you’ll be able to see the changes on the computer.

Probably the end of this class

Collaborating with other people

1. Make sure everyone has a Github account
2. Login to Github
3. Tell everyone to go to <https://github.com/BMSS-Programming-Club/Git-Lesson>
4. Show how to work in collaboration with other people
5. This is the repository we’ll be using. You can only look at it. If you want to make changes, you have to click fork. This will create a copy in your Github account
6. It’ll tell you it’s forking. Click refresh or wait a few seconds
7. Now we have a carbon copy of this repository in our own github account.
8. Let’s clone this in order to be able to make changes to it.
9. Click the clone or download button (it’s green)
10. Copy the link
11. Open GitKraken>Click the top left folder>Clone
12. Choose a filepath where you’ll clone this and paste the url. Click clone the repo!
13. Click open now at the top
14. Create a new branch called add-name
15. Let’s make some changes to README.md
16. Open it and just write your name on the second line.
17. Save your changes and commit in GitKraken
18. Push it to Github.
19. In Github, we can directly do a pull request here.
20. Click compare and pull request
21. Click create pull request
22. It’ll check to see if it can automatically merge. If you aren’t the first one, it should tell you there is a conflict.
23. Since so many people wrote on line 2, there is a merge conflict. It can’t merge automatically, we need to do the computers work for it
24. Click resolve conflicts
25. It should show you something like this 🡪
26. It is showing you what is in your add-name branch and what is in the master branch.
27. We want to keep both things. In this scenario, all we want to do is delete all the things that look foreign to us. In this example it would be the add-name-2 and master and all the “>”, “<”, and “=”
28. Now click mark as resolved and click commit merge. Sometimes you have to step 27 twice for some reason.
29. Click Merge pull request. If that button isn’t there, repeat step 27 again. It’s a bug with Github. Click Confirm merge to merge it
30. Now your name should be on the github page.