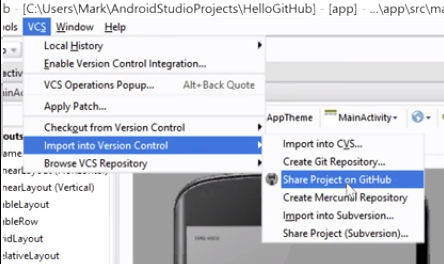
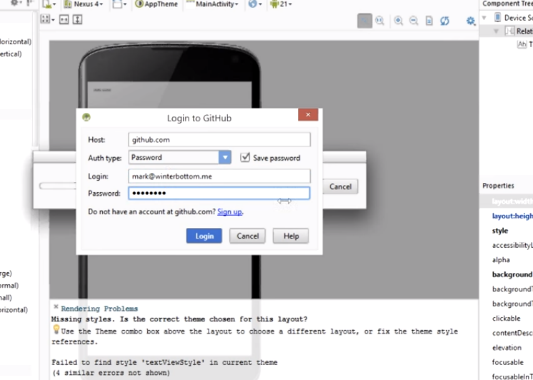
<https://www.youtube.com/watch?v=TU6XB9UdoMU>

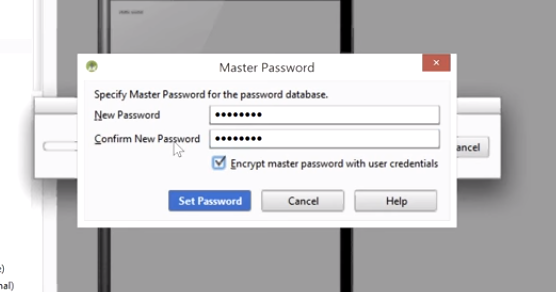
**How to push and Android Studio 1.x project to GitHub**

1. Go to VCS -> import into version control -> Share project on github

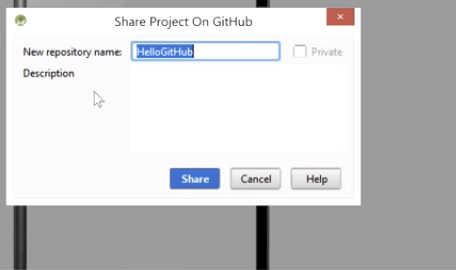


2. Sing in to your github account if it is the first time.

3. set a password for the database (could be the Fibonacci sequence). Don’t forget **to Check** the check mark, and click save password.

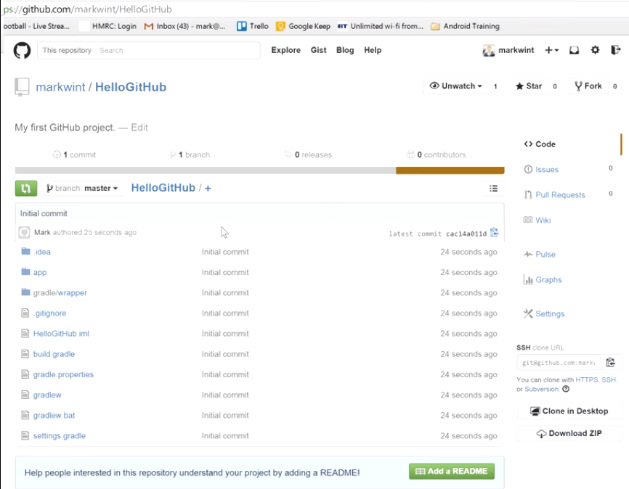


4. Put the title and a description of your repository, then click share.



5. Make a initial commit on github, just press the button commit.

6. Your new repository on gitHub



<https://try.github.io/levels/1/challenges/1>

**GIT BASIS**

Git is a version control system, that allows you to have multiple projects all changing at the same time but without losing track of the previous ones.

1. First find the git shell, it is just like a windows shell, so you can change folders an do your same shit. cd -> go directory, cd.. -> go back, dir -> stuff in folder.

2. To start an **empty** repository on your folder (can be any folder) type

*1. git init*

3. Now you can view the status by tiping :

2. *git status*

NOTE: Git status has four clasifications, and the problems with android studio come from here.

// this is ok, just commit (git commit -m "Add cute octocat story")

- staged: Files are ready to be committed.

// solve: add the changes to git(git add octocat.txt)

- unstaged: Files with changes that have not been prepared to be committed.

// solve: you need to stage the file that you’ve created (git add octocat.txt)

- untracked: Files aren't tracked by Git yet. This usually indicates a newly created file.

// solve: removing file from staging area (git reset octocat.txt)

- deleted: File has been deleted and is waiting to be removed from Git.

4. Say you **added** a folder or made **changes** to it, either way you need to put it in the staging area, so that it can be committed.

3. *git add <filename> or git add <wildcard>*

3. git add octocat.txt : adds ocotcat to the stage area.

3. git add “.txt” : adds every text file.

5. Notice how Git says changes to be committed? The files are in the Staging Area, and they are not in our repository yet. We could add or remove files from the stage before we store them in the repository.To store our staged changes we run the commit command :

4. *git commit -m "Add cute octocat story"*

6. To see the history of your commits:

5. *git log*

7. We’ve created a new empty GitHub repository at https://github.com/try-git/try\_git.git. To push our local *repo* to the GitHub server we'll need to add a remote repository. This command takes a remote name and a repository URL.

6. *git remote add <RemoteName> <location>*

6. *git remote add origin https://github.com/try-git/try\_git.git*

NOTE: this lines just sets origin to be the external repository, the goal is the **save the changes** of our local repo to that. Our external Repository **can be named anything** but it’s normal to call it origin.

8. Finally, we want to push all our files to the external repo.

7. *git push -u origin master*

Origin: the repository we want to save our local repo.

Master: Our current local repo.