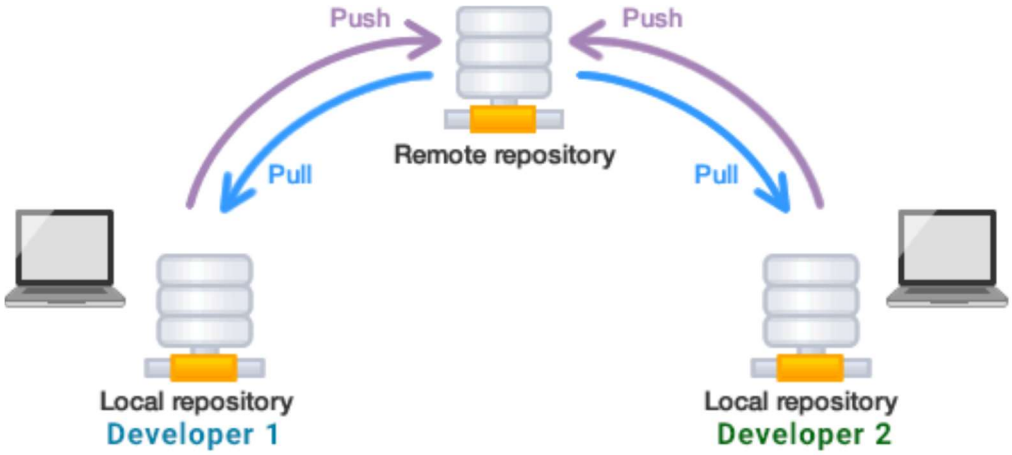
Introduction to Github

What is a remote repository?

Remote repositories are versions of your project that are hosted on the Internet or network somewhere.

Collaborating with other developers involves managing these remote repositories and **pushing** and **pulling** data to and from them when you need to share work.

To be able to collaborate on any Git project, you need to know how to manage your **remote repositories**.



What is Github?

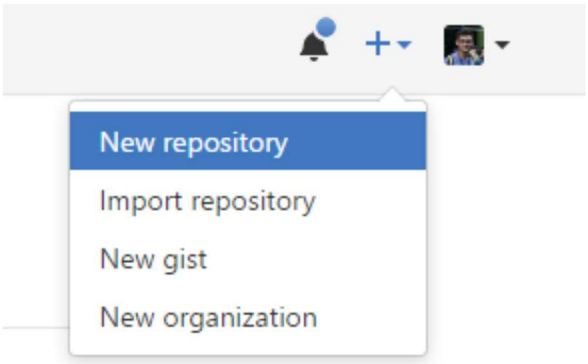
GitHub is the single largest host for Git repositories, and is the central point of collaboration for millions of developers and projects.

A large percentage of all Git repositories are hosted on GitHub, and many open-source projects use it for Git hosting, issue tracking, code review, and other things. So while it's not a direct part of the Git open source project, there's a good chance that you'll want or need to interact with GitHub at some point while using Git professionally.

Creating a new repository on Github

1. To create a new repo on GitHub, **log in** and go to the GitHub home page.

2. Click the plus symbol in the upper right-hand corner, and click **New repository** from the dropdown. There you'll see the new project creation form.



3. In the **Repository name** field, add a name. For the current task, name it "Task-X".

4. Give it a **description** if you want. Perhaps "Github Project for Task 0".

5. Then, leave the project with the default of **public**. That way anyone can find it if they search for it.

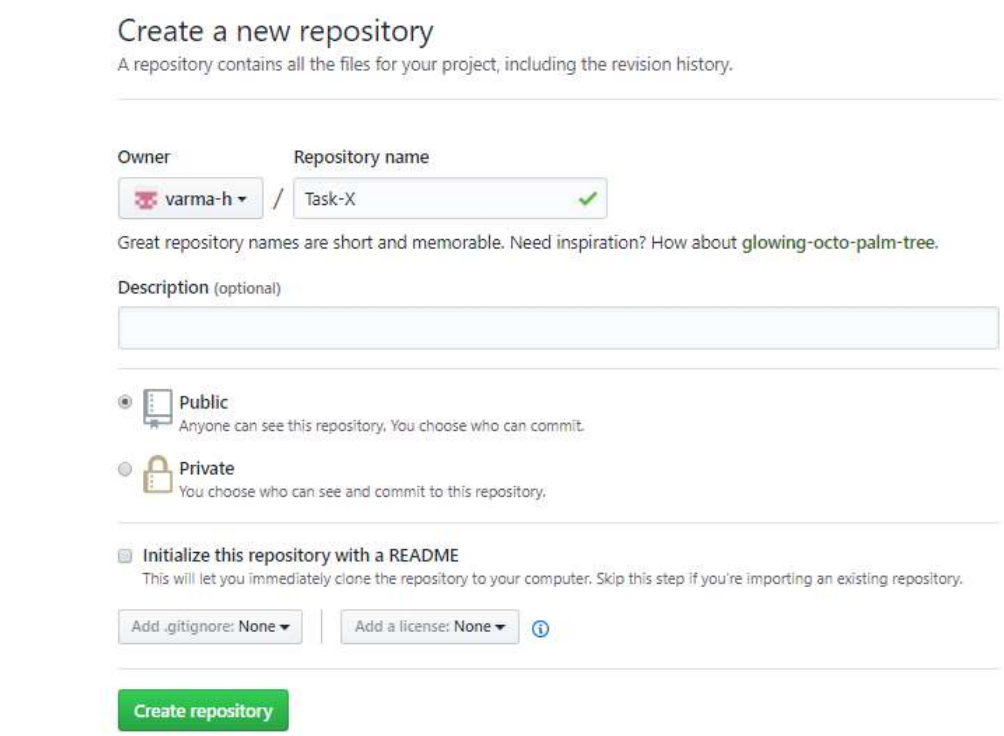
6. Leave the **Initialize this repository with a README** checkbox as it is, i.e., **unchecked**.

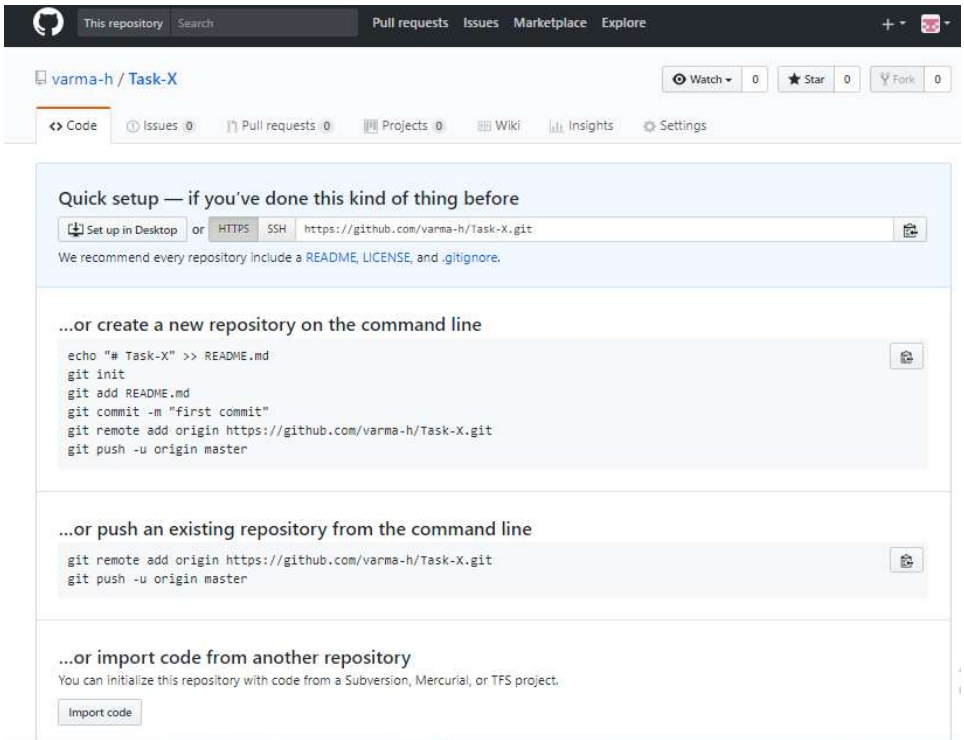
7. Finally, leave the two select boxes set to **None**. Now click on **Create repository**.

8. You'll now be taken to the **quick setup** page.

9. We're going to add this GitHub repository as a **remote** for our project. To do that, **copy the link of the repository** provided.

It should be something like: https://github.com/username/Task-X.git





Adding your Github repository as a remote in Git

We'll now link our local repository with our Github repository. After copying the link of your Github repository, follow the given steps:

1. Open **Terminal** in Ubuntu/Mac or **Git Bash** in Windows. Change the directory to your working directory, i.e. the directory containing your Git repository (which you wish to **push** to Github).

2. For example, in the previous tutorial, our working directory was **Desktop/Demo/**

Switch to this directory, and check the log:

$ cd Desktop/Demo

$ git log

commit 99a1e76a7524f19141916790b6d60551be18a72c

Author: username <email>
Date: Sun Dec 11 14:57:49 2016 +0530

Demo.txt file created

3. Now, we will add our Github repository as remote. For this, we use the **git remote add origin** command.
Something like: **git remote add origin <repository link>**

$ git remote add origin https://github.com/username/Task-X.git

Here the repository link is the **link we copied in the previous section**.

4. We can check if our remote has been added, by using the **git remote -v** command:

$ git remote -v
origin https://github.com/username/Task-X.git (fetch)
origin https://github.com/username/Task-X.git (push)

5. If you wish to change the remote url in the future, you can use:

git remote set-url origin <new link>

$ git remote set-url origin https://github.com/username/Task-X.git

6. Now, we will push our project to the remote repository. For that, we use **git push origin master**.

$ git push origin master

(Git might ask you for your Github username and password)

Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 278 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/username/Task-X.git
** [new branch] master -> master*

7. We have now successfully **pushed** our project to Github. We can open our Github repository to check.

8. Now, if suppose a collaborator **makes some changes** to the same project, and **pushes** his changes to your Github repository.

If you want to ensure that your local repository is **up-to-date**, you should use **git pull origin master**

$ git pull origin master

9. If suppose, you do not have your project saved locally, but have it online as a Github repository, you can make a copy of it on your system using the **git clone** command.

$ git clone https://github.com/username/Task-X.git

This will create a **local Git repository**, which is a **copy of the Github repository** you just cloned. This repository will be created in the **present working directory**.

(Suppose if you change or format your computer. You can clone your repository and continue working on it from your new computer.

Another case would be if you wish to collaborate to an already existing project on Github. The first thing you would want is a local copy of the project. For this, you will use the git clone command.)

That's all in the Introduction to Git and Github. You would be required to be familiar with these basics for the Winter Project.

You can also get more familiar with Github by getting to know about **forking repositories**, **starring/watching them**, **adding collaborators**, etc. You can also add **README.md** and **.gitignore** files to your Github repository (look them up, if you wish). These aren't necessary for now.

For Git, some advanced commands include commands for creating a new **branch**, merging two branches, and many more. Again, this isn't necessary for now, as you won't be needing these commands for this Winter Project.

Every developer is expected to know these basics.

Henceforth, the method of submission for each task would be pushing your project to Github.


iecSe
APPDEV
WINTER PROJECT
2017-18