* This document gives the basic procedure for creating and updating a local Git repository, called a remote repository, by creating a clone of one of the repositories on GitHub.
* This document assumes that you have installed Git on the computer that you want to use as your remote repository. If you have not installed Git yet, please see the document named Installing Git included in this folder.
* There are two approaches to creating a local repository:
  + the first is to initialize a repository in an existing directory on your computer,
  + The second is to clone a Git repository from another server.

**Initializing a Git Repository in an Existing Directory**

1. Open Git Shell and change directory to the location of the project that is to be tracked
   1. $ cd your\project
2. Initialize Git in that project’s directory
   1. $ git init
3. If the init was successful, there will be a colorful, bracketed message next to the directory that you’re in. This message informs you about the status of your repository. Currently, Git is not tracking any of the files in this directory, so tell it to track all of the files in the directory:
   1. $ git add --all (Note: in front of all there are two dashes, ‘-‘)

Or

* 1. $ git add –A (Here there is only one dash)

1. Now that Git is tracking all of the files and folders within this directory, we can commit them
   1. $ git commit –m “A descriptive message of what was committed”
      1. Note: if there are changes added to Git and git commit is entered, the Git shell will open a word editor for the user to enter a message. We skip this by issuing the “-m” flag to git commit.

At this point, all of the files that existed in your directory are now being tracked by Git and your remote repository has been created.

**Cloning an Existing Git Repository**

To clone the entire Git repository on your local computer the steps are similar

1. Open Git Shell and change directory to the location for the clone
   1. $ cd your\cloned\project
2. Initialize Git in that project’s directory
   1. $ git init
3. Add a link to the external Git repository and name it “origin”
   1. $ git remote add origin [url]
      1. [url] is the link to the repository you want to clone
4. Bring all of the files from the “origin” repository into your “master” repository
   1. $ git pull origin master

The remote repository is now set up and contains the complete information of the external repository to which it is linked.

**Pulling and Pushing Updates between Repositories**

After cloning a repository or adding ‘origin’ as a remote to our repository, we may push and pull between our repository (master) and the external repository (origin). Pulling means we are going to fetch down any changes that were made to origin since we last checked and pushing means we are sending our local changes to origin to be stored.

**Preparing the Repository to Push or Pull**

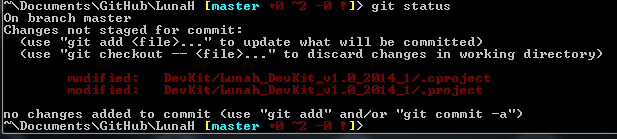
Before sending a push or pull command, it is important to run a check on what is currently in master. Do this by running:

1. $ git status

This will display any changes that your local Git has tracked. If there is nothing to report, you will see something like this:



If, instead, you see something like this:



Then before pushing or pulling, these changes must be committed. The first step is to add the changes so that Git knows we want to keep them:

1. $ git add --all

On a successful add, the numbers next to master will turn green to indicate that Git is tracking the new changes. Once the changes are added, we may commit them to the repository:

1. $ git commit –m “message about the changes”



There are no error messages, so our commit has been accepted. To double check that our commit and message are stored, check the log:

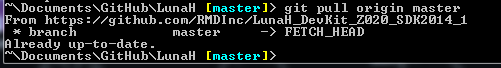
1. $ git log

From this point, we may push or pull changes to origin because our side is up to date.

**Pulling from Origin**

The command to pull changes is quick and gives the result of what was pulled from origin:

1. $ git pull origin master



Since there were no updates to pull down from origin, Git tells us that our repository is up-to-date.