**Introduction**

Git allows us to keep track of versions of code as we work on it. A repository is essentially a directory that acts as the source for our project code. Like any file system, our git repositories can have anything from java files to images. However, unlike a typical file system, a git repository has a series of snapshots associated with it, essentially saving the state of the directory and all of the files in it. There are two types of repositories. We can use git to create local repositories on our own machines. We can also have a remote repository on GitHub (git != GitHub). In our local repository, we can pull, or retrieve changes, from a shared remote repository. Conversely, we can push, or publish, changes we make locally to the remote repository.

**Tutorials/Resources**

1. Start off watching this video for a high level overview

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

1. This tutorial walks through and explains some basic commands

<https://www.youtube.com/watch?v=SWYqp7iY_Tc>

1. Give some of these commands a try

<https://learngitbranching.js.org/>

**Typical Project Git-Flow**

The way that git will be used in project 2 and 3 is more typical to that of agile software development practices. In that case, there is the master branch, managing the production version of an application. A development branch is made off of the master branch, and issue branches, or feature branches, are made off of development. Each branch has a copy of the source code, but this setup allows for each feature to be developed and tested independently so they may be done concurrently. When a feature is completed, its branch is integrated back into dev, and ultimately dev will be integrated into master. There are often branches between dev and master as well to accomodate for staging and testing environments.

In the case of our batch-source repository, however, we are not following that convention. Instead, we have branches that will not have common code but have unrelated commit histories. Each branch will serve as a place for you to keep version history of your exercises and assignments.

The following tutorial will walk you through creating a local repository with a branch for your work, that will correspond to the remote GitHub repository batch-source.

**Setting up your own branch in our batch-source repository**

1. Clone remote github repository in an empty folder

> git clone https://github.com/1901-Jan14-Spark/batch-source.git

2. Navigate into the batch-source folder, which should appear wherever you cloned the repository.

   Use the cd command to change directory

> cd batch-source

will move into the batch-source folder if you were one folder up in the file directory

> cd ..

   will navigate you one folder up in the file directory

Steps 3. and 4. need not be completed. Cloning a remote repository will automatically initialize a local repo with the remote “origin” set to the remote repository cloned from. (skip to step 5)

3. Create a local repository within the batch-source folder

> git init

4. Store the location of the remote repository in the directory where your local repository is.

> git remote add origin https://github.com/1808-Aug13-Java/batch-source.git

   This will save a remote name of origin associated with the remote url

5. Create a branch in your local repository with your first and last name

> git checkout -b First\_Last

6. create a branch in the remote repository (this will be in the repository at GitHub.com) with the same name as your local branch

-- You are ready to push your commits to the remote repository --

To commit your code locally:

1. First you must add your changes to the staging area git

> git add -A

   This will add all of your changes

> git add file.txt

   This will add file.txt to the staging area

\*\* to see what changes are present and which have been added to staging at any time, the command `git status` will show you

2. Committing your code after this will commit the changes from everything in the staging area

> git commit -m "A concise but descriptive description of your commit goes here"

To push your commits to your remote repository

> git push origin First\_Last

* Push at least one file to your branch by the end of the day (can just be a test txt file)