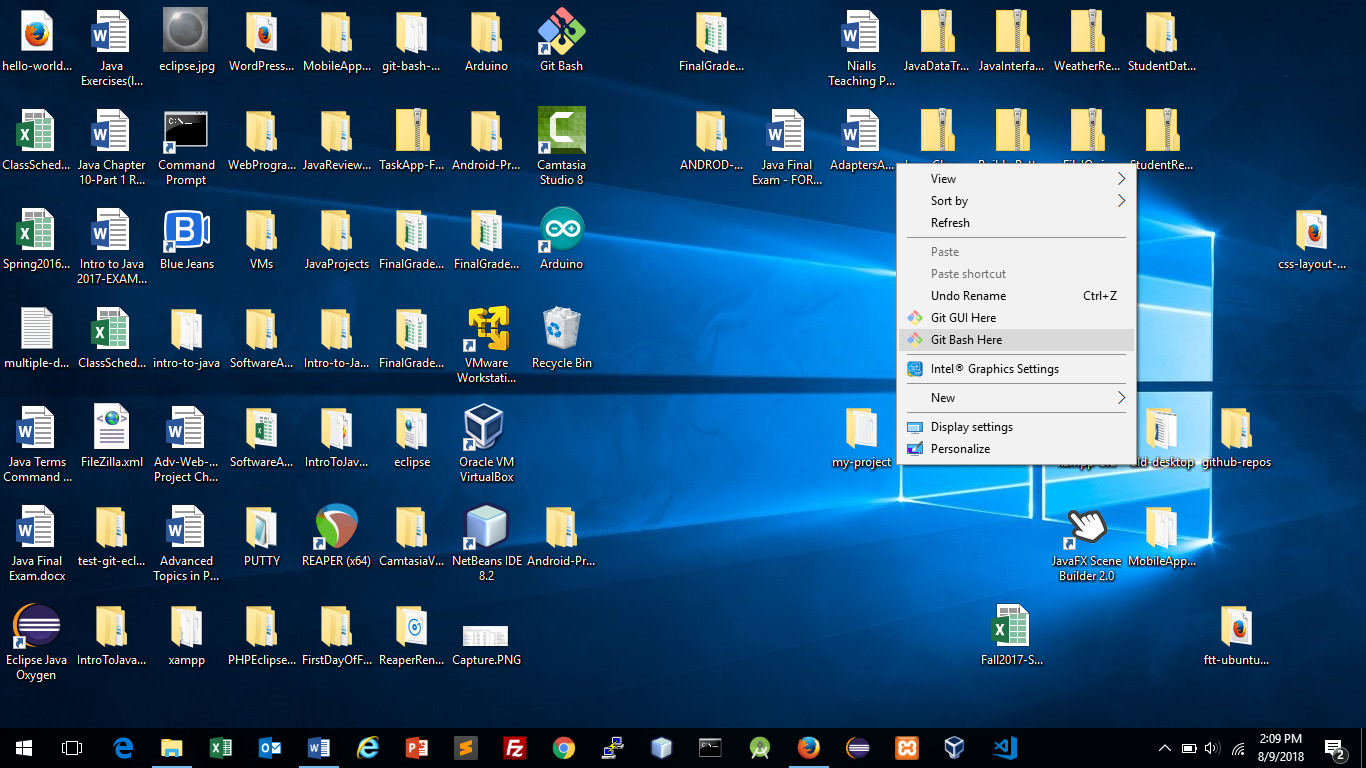
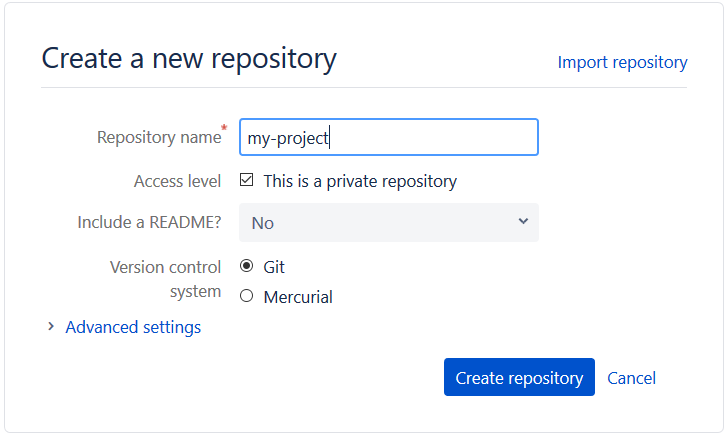
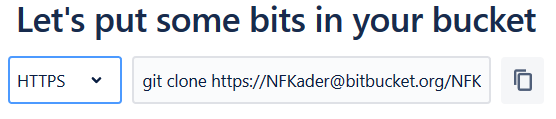
**Incorporating Git and BitBucket Into a Project**

1. First, set up an account with BitBucket. You can sign up at this url:
   1. <https://bitbucket.org>
2. Install **GitBash** for Windows (this is the command line tool that allows you to interact with your git repositories and keep them synched with BitBucket).
   1. [Here is the link to download Git Bash for Windows](https://git-for-windows.github.io/)
   2. The installation is very straight forward, you can just choose all the default settings
   3. **Note that GitBash uses Linux commands** (which are different than the commands you would use on the Windows command line. So it will help you greatly to learn the very basic commands in Linux.
3. For the next few steps, we’ll be using a dummy project named **my-project**. Download the .zip file, extract it, and put the my-project folder on your desktop. The project consists of a single file named index.html.
4. There are a few different ways that you can set up a project to use Git/BitBucket. In this approach, we’ll create a local repository first (on your local computer). Then we’ll create a remote repository on BitBucket. Then we will link the two repositories so that they effectively become one.
5. Right click on the my-project folder and choose Git Bash here…



1. The Git Bash terminal will appear and it will automatically navigate to the my-project folder.
2. Create a new local repository by entering this command:

**git init**

1. Leave the Git Bash terminal open, we’ll come back to it in a minute. Next we’ll create the remote repository on BitBucket.
2. Open a browser and log in to bitbucket.org.
3. After you log in, click the plus sign in the left hand side of the screen, Then click on the **Repository** link.
4. You should see a screen that looks like this:
5. In the Repository Name field, enter **my-project**.
6. Click the blue **Create Repository** button.
7. You have now created the remote repository. So next we’ll link the local one to it.
8. Make sure that you select **HTTPS** from the drop down list:
9. Now copy the url for the repository. Note that you do NOT need to include the ‘git clone’ that appears before the url. So select everything starting with https:// to the end of the url.
10. Go back to the Git Bash terminal and enter this command (you’ll have to paste the url at the end of the command, so replace the x’s with your repositories url):

**git remote add origin** xxxxxxxxx

1. The local and remote repositories are now linked.
2. To add the project files to the local git repository, you must ‘stage’ them first. Enter this command to add all the files to ‘staging’:

**git add .**

1. Your next commit (which we are about to do) will commit all the files in ‘staging’ to your local repository. Enter this command to commit them:

**git commit –m “First commit”**

1. Finally, you’ll want to ‘push’ the files to the remote repository on BitBucket. Then the local repository and the remote repository will be synched. To push the latest commits, enter this command:

**git push origin master**

1. Go back to the browser and click on the **Source** link on the home page for your repository in BitBucket. You should see the source code files for your project now (in this case, just index.html). Your local and remote repositories are now synced.

For more information about Git and BitBucket, watch this video:

<https://youtu.be/NV8zbr4Md5w>

**Using Git on a Daily Basis**

It’s really important to make Git a part of your regular programming routine. Any time you make changes to files in your project you need to ‘stage’ and then commit the changes. All the files added to ‘staging’ will be included in your next commit. To ‘stage’ all files that have changed since your last commit enter this command in the Git Bash terminal:

**git add .**

Note that the dot indicates that you want to ‘stage’ all files that have changed. If you want to add a single file only, you can replace the dot with the path to the file name.

Once you have staged all the files that you want to include in your next commit, you can go ahead and run this command:

**git commit –m “Some message goes here”**

The ‘m’ option stand for message. Every commit should include a descriptive message.

It’s not unusual to make multiple commits during a single programming session. And there are strategies for organizing the files in each commit.

When you are done with your programming session, you should always ‘commit and push’. Make sure that you commit all of your changes, as has just been described, then ‘push’ all of the commits to the remote repository (on BitBucket). That way, if your harddrive crashes, the latest changes to your project are safe in the cloud (the number one rule of programming is to always make sure your files are backed up!). To push your files to BitBucket, enter this command:

**git push origin master**

We’ll learn more about ‘origin’ and ‘master’ later on. We’ll also see a few other git commands that you need to get comfortable with in the next activity.

**Activity: Working on my-project**

Right now, my-project consists of a single html file. Let’s make some changes to the project.

* 1. Open index.html in your IDE and make a change to it (you can change the text inside the H1 tag).
  2. Add another file to the my-project folder, name it **test.html**
  3. Put some simple HTML in the test.html file.
  4. In the Git Bash terminal, enter this command:

**git status**

The git status command will show you all the files in the project that have changed since your last commit. Notice that the output from running this command is colored red. This indicates that they have not been staged, and will therefore not be included in the next commit. Also note that the files are organized into two separate groups; those that have already been commited, and those that are new and have not yet been committed.

* 1. Enter this command to see the changes that have been made to index.html since the last commit:

**git diff index.html**

* 1. Add the files to staging by running this command:

**git add .**

* 1. Now go ahead and run the **git status** command again. Notice that the files are now listed in green, which means that tat are in ‘staging’ and will be included in the next commit (notice the messeage that says ‘Changes to be committed’).
  2. Commit the changes by running this command:

**git commit –m “Updated index.html and added test.html”**

* 1. Run this command to see a list of commits for the project (note that the commits are ordered so that the most recent ones appear first):

**git log**

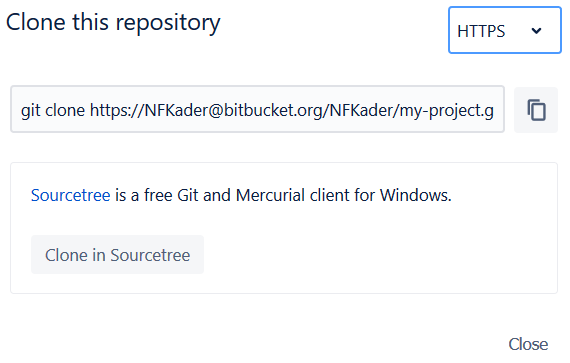
* 1. Now push the changes to BitBucket, so that the local and remote repositories are in sync. Run this comand (you may have to enter your BitBucket username and password):

**git push origin master**

* 1. Verify the that the remote repo has been synched by refreshing the Source page in BitBucket. You should see both index.html and test.html
  2. Click on the Commits link in BitBucket to see a log of commits.

**Cloning a Repository to other Computers**

You can ‘clone’ your remote repository to your home computer (so that you can work from home, as well as at school!). To do so, follow these steps.

* 1. Make sure that you have Git Bash installed on your home computer.
  2. Create a folder called **repos** that can hold all of your projects (for Advanced Web Dev, we’ll put all of our projects inside the htdocs directory of your XAMPP installation, but for this activity you can use any folder since we are just experimenting with Git and BitBucket).
  3. Right click on the repos folder and choose **Git Bash Here**. This will open Git Bash and navigate directly to your repos folder.
  4. You’ll need the url for your repository on BitBucket. So login to BitBucket and then click on the **my-project** repository.
  5. On the home page for the repository, you’ll see a button labelled **Clone** on the right hand side. Click that button and you should see a pop up window that looks like this:
  6. Make sure that the drop down list on top is set to HTTPS.
  7. Now copy the git clone command and the url for your repository. To do this you can simply click on the copy icon. This will copy the git clone command that you’ll need to run in Git Bash.
  8. Go back to Git Bash and paste the command into the terminal, then press Enter to execute it. This will ‘clone’ the repository to your repos folder. You maybe be prompted to enter your BitBucket password, and after you enter it, you should see a folder named my-project appear inside your repos folder.
  9. Now you have 3 versions of the repository. The one on BitBucket is known as the ‘origin’. Whenever you work on your project you should always do these two things:
     1. Before you start working, pull all of your changes from the origin repository (on BitBucket) by running this command:

**git pull origin master**

This will make sure that your local repository has all the latest commits/changes that have been pushed to the origin.

* + 1. When you are done working, commit all of your changes and then push all of your commits to the origin repo by running this command:

**git push origin master**

It’s very important to commit and push all of your changes when you are done with a programming session, otherwise you’ll run into problems. For example, let’s say that you work on the project on your home computer, but forget to commit and push your changes. When you come to class to work on your project, you won’t be able to pull the latest changes to your school computer. So always remember to PULL BEFORE YOU START CODING AND PUSH WHEN YOU ARE DONE!