Git Some Practice

**Create a local Repo**

1. Download git (www.git-scm.com)
2. Create a folder in your local computer which will hold your local git repo. Name it **GitPractice**
3. Open a command line window type **cmd** in the Windows search bar
4. Initial the **GitPractice** directory to put it under source control **init git**
5. Verify that the **GitPractice** folder now contains a .git folder. This ensures that git is now versioning your code **dir /a**
6. Create an ***index.html*** page and save it in the **GitPractice** folder you created
   1. include an *<h1>Spring has Spring </h1>*
7. Create a text file called ***README.md*** (make sure that the extension ends in the .md ending) Save the file in the **GitPractice** folder
   * 1. Type ***#Gitting Some Practice*** on the first line
     2. Type ***##Your Name*** on the second line

**Create a REPO in gitHub**

1. Open a browser and navigate to your gitHub account.
2. Create a new repo and call it **Spring**
3. Copy the link to the gitHub repo

**Connect your local Repo to your gitHub Repo**

1. In your local **GitPractice** folder make the connection to the Remote Repo **git remote add <paste the URL of your remote repo here>**
2. to check if the remote worked type **git remote -v**
3. Stage your files to prepare them for a snapshot in time **git add .**
4. Commit your files, preserving the snapshot of your code at this moment in time

**git commit -m “Initial Commit”**

1. Push your committed snapshot up to your remote repo (in gitHub) **git push origin master**
2. Check in your remote directory to ensure the files were uploaded to your remote repo

**Make some changes to your local files and push the changes to remote repo**

1. Create a style sheet for your index.html file. Save it in **GitPractice** folder.
   1. Add a style that makes the background of the index.html blue
2. Make some changes within the index.html page
   1. include a link for the stylesheet
   2. add a paragraph with some text that is at least two sentences long
3. **add**, **commit** and **push** (repeat steps 13 – 17)
   1. use “added a paragraph and a style sheet” for the commit message

**Make some more changes to your local files and push the changes to remote repo**

1. Make changes to the index.html file
   1. delete the paragraph
   2. add a “Springy” image
2. **add**, **commit** and **push** (repeat steps 13 – 17)
   1. use “replaced paragraph with an image” for the commit message
3. Make changes to your index.html file
   1. Connect your page to a bootstrap cdn
   2. Add a jumbotron
   3. Add a table and use bootstrap classes for styling
4. **add**, **commit** and **push** (repeat steps 13 – 17)
   1. use “Added Bootstrap” for the commit message

Turn in your assignment

1. In canvas, paste the URL link to your github repo. It should look something like: <https://github.com/MarkleyClassroom/Spring>

**Helpful Git Commands >>** <https://git-scm.com/docs>

|  |  |
| --- | --- |
| git init | Initialize a repository |
| git status | Displays the status of the files in the directory. Some examples:  **nothing to commit** (no new files/changes/deletions detected)  **untracked files** (files that are not staged yet)  **changes to be committed** staged waiting to be committed |
| git diff | Shows the difference in a file between versions |
| git add . | Add files to the staging area – waiting to be committed |
| git commit -m “message” | Commit the files to save this moment in time.  -m a message follows  “ “ put in a short description of the changes made in this snapshot |
| git log | Displays a timeline of committed files |

**Terminology**

|  |  |
| --- | --- |
| repository (repo) | A bucket or location where our files and versions (history) of the project are stored |
| branch | The current “street” of files you are working on |
| master | Usually the name of the branch that contains the latest working set of files |
| origin | The branch name generally given to the remote repo |
| README.me | A file that should be created when you initialize a directory. It describes the repo and/or provides documentation to the software.  # Heading Info ## Smaller text |