**Process Documentation for Git and Github Basic Functions Revised**

**Gabriel Linski**

**https://github.com/Gabe-Epsilon/TWS-Repo/tree/main**

**Installing Git on Windows**

1. Open a web browser and search for the git homepage or click this link: [https://git-scm.com](https://git-scm.com/)
2. Download the latest release for Git. Once the download is complete double click the installer.
3. Continue through the installer, no changes are required for our purposes.
4. When the installation is complete launch Gitbash.

**Creating a GitHub Account**

1. Open a web browser and search for the github homepage, or click this link, [https://github.com](https://github.com/)
2. Click on sign-up in the top right corner. Input your email, create a password and choose a username for your Github account. After this you will be required to verify your account with a code sent to your email address provided.
3. You may choose to personalize your account, or skip this for now.
4. Once your account is setup, click on the green button on the top left called Create repository. Name your repository, give it a description and now you will have an empty repository to store and work on files from.

**Creating a Local Repository with Git**

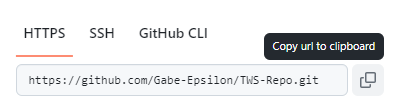
1. Open Git Bash. Determine your current file path by typing and executing: pwd.
2. To navigate to another file branch, type: *cd path*
3. To create a new project folder type: *mkdir gitproject*
4. Create a textfile by typing: *touch exercises.txt*
5. For changes to a file to be tracked, the file must be added to the staging area.
6. To do this type: *git add exercises.txt*
7. Anytime the file is changed it will be needed to be added to the staging area again.
8. To commit these changes type: *git commit -m "Message Here"*
9. Always include a short message about the changes you have made to the file for future reference and for others to know about the edits.

**Reverting to a Previous Commit In Git**

1. To revert to a previous version of our document, we first need to find the version we want from.
2. Type: *git log*
3. This will display a list of previous commits alongside their identification number and edit messages. Copy the id of the commit you want by right clicking as the keyboard shortcuts will not work. Press q to exit the log.
4. Type: *git revert id*
5. This will revert the file back to a previous commit.

**Adding Files to your GitHub Account**

1. Login to your GitHub Account and navigate to the repository you want to work from.
2. Click the Code button and copy the link to your repository.



1. The link should look something like "<https://github.com/Gabe-Epsilon/TWS-Repo.git>".
2. Open your Git Bash terminal and type: *git remote add origin link*
3. This will create a link between a remote repository from which you can push file commits to the GitHub repository.

**Branches**

1. To work on a branch, which is a copy of the main files, we first need to create it.
2. Our branch will be called edits. Type: *git branch edits*
3. To see all branches type: *git branch*
4. Switch to this branch by typing: *git switch edits*



1. Change and add files to your directory, these changes will be made to the branch we are currently in.
2. To track all files, type: *git a .*
3. To see the changes in git type: *git status*
4. The changes to the files will be listed in the terminal.
5. Commit all these files by typing: *git commit -a -m "Message"*
6. This will commit the change to this branch, not to the master branch.
7. Switch to the master branch by typing: *git switch master*
8. Then we will need to merge the changes from the edit branch into master.
9. Type: *git merge -m Message" edits*

**Pushing File Changes to GitHub from Git Terminal**

1. Since we have already created a link from our local repository to our GitHub repository, we can now push changes made to our files from the Git terminal.
2. First specify your main branch, type: *branch -M master*
3. To then push changes we've made from our local repository type: *git push -u origin master*
4. The changes we made are now synchronized to GitHub.
5. If we have multiple branches to push, type: *git push --all*