**GITHUB COMMANDS**

From the Session of Bazlur Rashid Vai.

* **Installing Git Bash:**

1. Go to Google and Search for “Git Bash Download”
2. On the Download page click the Windows Installer and on next page select the right kind of installer (Preferably x64 version if your PC is 64 bit, x86 if it’s 32 bit)
3. Click ‘next’ in public license part
4. Now there’s a page for options you can check or uncheck. If you are not being specific about any of the options, click next.
5. Select an editor of your choice in the next page or leave it as it is.
6. Next, you can either choose to use default branch name set by git or your own choice. Choose your preferred option and click next.
7. Next, there is a recommended option for using Git from command line as well as third party software. Unless you’re being specific, leave it as it is.
8. Next you can choose to use the internal SSH or install your own and let git use that path.
9. Choose an HTTPS transport backend.
10. Configure line ending conversions. I left it to git to decide.
11. Configure which terminal to use with Git Bash, I left this to git (MinTTY) to choose.
12. You can choose default git pull behavior. I left it to what was selected before.
13. Choose credential helper and configure extra options on next page. I left these to git to select.
14. You can either choose or not choose the experimental options next.
15. Later, it will show you the components to be installed. Click Install and it will install it.

* **Cloning A Repository:**

1. **Generating SSH Key:**
   1. Open Git Bash, put this command:

**git clone git@github.com:bazlur36/git-basics.git my-git-project**

This will prompt that your key will saved in user/username/.ssh folder. Press **Enter** if you’re comfortable with the directory. May give you a prompt for passcode but avoid it.

* 1. Go to the directory where the Key is saved and open the **.pub** file in a Notepad. Copy the key.
  2. Go to your Github Profile 🡪 Settings 🡪 SSH and GPG Keys 🡪 New SSH Key 🡪 Paste The Key and Set A Name for the key.
  3. Now you can copy the **SSH Link** of a Project in Github and Clone It.

1. **Cloning:**
   1. Open Git Bash in the Directory where you want the project to be and write the command below:

**git clone “SSH Link” “A\_Folder\_Name\_Of\_Your\_Choice\_If\_You\_Want”**

This will clone the entire project into the directory of your choice.

* 1. **cd “The\_Folder\_Name\_You\_Chose”** move to the folder of project.

Use this command to see any changes in files, what has been committed and what is waiting to be committed:

**git status**

* **Creating or Moving To New Branch:**

1. Creating branch:

**git checkout –b “branch\_name”**

1. Moving to a branch:

**git checkout “branch\_name”**

* **Checking for Changes In Edited File:**

If you want to see the difference between the code in the file before and after you changed it, use the command below:

**git diff “file\_name”**

* **Staging the File For Commit:**

Ready the file for commit and add it to the tree:

**git add “file\_name”**

* **Committing The File To The Local Repo:**

**git commit –m “Message”**

**git branch –a :** Shows all of the Branches and adds a star next to the branch you are in, also show which of them are remote.

**git push origin “branch\_name” :** Pushes the changes into the branch you chose.

**git fetch –-all :** Since you only have information about your own Branch and the Main Branch (If you have pulled from Remote Already [We’ll See Later]), you can’t see other branches other contributors have created. **This command fetches all of the Branches**.

**git merge origin main :** Using this command, you can see if you have any update missing to the **main** branch.

**git pull origin main :** This command is used to basically update your local repo with the current remote repo [The Name Pull Suggests that]. Conflict: If you made changes in a file, and another contributor has also made some changes to that file, let’s say reviewer has merged the Other Contributor’s changes to the main branch. Now if you make a Pull here, it will return a Conflict. If there’s a conflict in the file, the file will look like this:

This is the first line of code in the file.

<<<<<<< HEAD

This is the second line of code added in the main branch. This is the third line of code added in the main branch.

=======

This is the second line of code added in the feature branch. This is the third line of code added in the feature branch.

>>>>>>> feature-branch This is the fourth line of code in the file.

The ‘=====’ separates the conflicting changes in that file showing that in that same line another piece of code or change has been made.

This conflict needs to be resolved manually alongside the other contributor.

After resolving you can do git add “file\_name” to add the file to the tree for commit, and later commit.

After this you can do a Push to the branch.

* **Creating A Pull Request**

On github, go to **Pull Request**. You’ll see an Editing Screen, this will indicate which branch you updated recently and that you are requesting for a merge to the main branch. On the right side you can also select the Reviewer. After that you can submit.

Reviewer can later comment on your changes and request new changes.

You can also see your commit history in this page.

A purple merged notification will appear in this page.

**For Detailed Information:** [**https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax**](https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax)