**Git Commands**

***git –version*** *– To check the version of git on your machine.*

***git config –global --list*** – To see your git user configuration details.

***git config –global user.name “someName”*** – To set your git username. To be used everywhere not just on your local machine hence the “--global”

***git config –global user.email “someEmail”*** – To set your git email address. To be used everywhere not just on your local machine hence the “--global”.

***File\_name.md***– mark down

***git clone*** [***git@github.com:Daniel-Karongo/First-Demo.git***](mailto:git@github.com:Daniel-Karongo/First-Demo.git) - Download a repository from git to your local machine to some folder.

***dir /a*** – Shows all the files and folders in a directory, including the hidden files. Windows CMD.

***Get-ChildItem -Force*** – Shows all the files and folders in a directory, including the hidden files. Windows Powershell.

***git*** – Store all the code changes to your repository/project over time

***git status*** – Shows all the files that were created, modified or deleted but that have not been saved in a commit yet.

* Untracked files – Files that git doesn’t know about (newly created files)

***git add <somefile.extension>*** - Tracks (stages) the file specified. E.g. *“git add index.html”*

***git add .*** – Tracks (stages) all the files in the repository ready for commiting.

***git rm –cached <somefile.extension>*** - To unstage the file specified. This works even if the file had been committed.

***git commit –m “Some change” –m “Some other change”*** – Commit the staged files but only globally, not live to say, github.

***git commit –am ‘some message’*** – Stages and commit the files at the same time, but it only works with modified files.

***git remote add origin*** [***git@github.com:Daniel-Karongo/First-Demo.git***](mailto:git@github.com:Daniel-Karongo/First-Demo.git) – To set up the push by first specifying where you want to push the repository. Used when you create the repository locally first, rather than cloning from github. You will however have to create an empty repository on github to get the link to copy.

***git branch –M main*** – Similar to the above but for setting up the *“main”* in *“git push origin main”*

***git remote –v*** – To check that the above has been set up.

***git push origin main*** – Saves the committed files to a remote repository, say, on github

***git push –u origin main*** *– To set up an upstream so that when you want to push later, you only need to type “git push”. Short hand for* ***“git push –set—upstream origin main”***

***echo “some text” >> someFile.extension***– To create a new file and push some text into the file. E.g. *echo “# Hello World” >> README.md*

***cat <somefile.extension>*** - Prints out the contents of the file passed. E.g. *“cat README.md”*

***pbcopy < <somefile>*** - Copies the contents of the file to the clipboard.

***git init*** – Create a new git folder within an uncloned repository. Makes the folder where this command is run to be the working directory.

***Git init –b main*** – Create a new git folder and makes the name of the main branch as main. Originally it was master, hence the command.

***git reset*** – Undoes the last staging (git add …) when the staging has not been committed. [unstages]

***git reset HEAD~1*** – Undoes the last commit [unstages]

***git reset –-hard b9e776e63d5e4ad3dd53df71de4e335ad795bb56*** – Permanently undoes/removes all the commits after the log passed

***git log*** – Shows all the previous commits in chronological order. Each commit has a unique checksum, 40 characters long, generated using SHA-1.

***git log –pretty=oneline*** – Similar to the above command but returns the commits’ checksums along with their messages on a single line.

***git tag –a <version number> -m <some message>*** - To add a tag to the project at the current commit. E.g., *git tag –a v1.0. -m “First release”.*

***git tag*** – To show all the tags you have given project until that time.

***git push origin v1.0.*** – To push the tags to the server. Done for each of the tags.

***git show <tag name>*** - To show more information about the tag specified. E.g., *“git show v1.0.”*

**SSH Keys**

Used when wanting to commit code to a remote repository like github to prove that you are the owner of the github account. Different from http in that you will not need to log in in order to push your changes to github.

Procedure:

* Generate a key locally. ***ssh-keygen –t rsa –b 4096 –C “***[***githubemailadderess@gmail.com***](mailto:githubemailadderess@gmail.com)***”***. –t – type of encryption. –b – Strength of encryption. –C – github account name.
* Give the key generated a name. It will be stored in the “User’s” directory under the “.ssh” directory. You can also assign a password to the key if you want.
* Two files will be created in the current directory in the git bash. One has an extension of “.pub”. This is the one you will upload to github interface. The other file without the “.pub” is the private key and should be kept securely in your local machine.
* Go to github then go to settings then to “SSH and GPG keys” then add a “new ssh key”.
* Make the local git CLI work with the key you just generated.
* Launch the ssh-agent: ***eval "$(ssh-agent -s)"***
* Add the ssh private key to the ssh-agent: ***ssh-add ~/.ssh/id\_ed25519***
* You will now be ready to push you files to github

**Branching**

***git branch –*** Returns all the branches associated with the current project

***git checkout –b ‘some branch name’*** - Leaves the default branch, creates a new one called whatever you call it and goes to that new branch. *git checkout* is an alternative to **git switch**. E.g. *“git switch –c branch2”* does the same thing as the command in bold.

***git checkout ‘some branch name’*** – Similar to the above but the branch passed is already existing.

***git switch -*** - Checkouts to the previous branch.

***git diff*** – To show the differences between the files in the working repository with those that were committed last.

***git diff*** –staged - To show the differences between the staged files and those committed last.

***git diff*** ‘some branch name’ – Shows the differences between two branches

***git pull origin main –*** Merges the local main branch with the remote main branch, i.e. pulls the changes from the server to the local machine. Done before merging the local main branch with another local branch so that you can then push the changes to the remote main branch.

***git branch*** – Shows all the local branches.

***git branch --all*** – Shows all the branches, both local and remote.

***git branch –d ‘some branch name’*** – Delete a branch once it has been merged with the main branch (mostly).

**git merge ‘some branch’** – Merges two branches, the one you are on, and the one you have passed. This takes the changes in the branch you are specifying and merges them to the branch you are on. E.g. If you are on “main”, “git merge Karongo” updates the “main” branch with the changes from “Karongo”