**BASICS OF GIT**

<https://www.youtube.com/watch?v=USjZcfj8yxE&t=164s>

<https://videotutorials.notion.site/Introduction-to-Git-ac396a0697704709a12b6a0e545db049#d5e9c2b6379246a593c1ef74051e7e3c>

**DOWNLOADING GIT**

1. <https://git-scm.com/>

**OPEN TERMINAL**

1. >> git –version

2. Configure name and email

>> git config --global user.name "Your Name"

>> git config --global user.email “[your@email.com](mailto:your@email.com)”

**3. REPOSITORIES**

Local – saved on computer

Remote – saved on a remote server

**Initialize a repository**

1. go to the directory

>> cd documents/coding

##create a new folder/directory

>> mkdir <folder\_name>

>> mkdir git\_intro ## example

>> ls #make sure you are inside the directory

**## INITIALISE REPO**

>> git init

>> ls .a #check out the git repo is created

#.a is to see even the hidden files inside the directory

**4. Check Status**

>> git status

**5. Create a new file**

##code is written already for these files

>> touch index.html

>> touch app.js

>> touch styles.css

**6. Staging Files**

>> git add file.js ## to add individual files

>> git add file1.js file2.js file3.js

Or to add all the files together

>> git add .

>> git add index.html #example

**7. COMMIT**

>>git commit -m “commit message”

>>git status

**8. GIT LOGS**

>> git log

**9. GIT BRANCHES**

>> git log

##generates a list of git commits that has been done

##to go back to a previous version

>> git checkout <commit-hash> #just paste the hash number from the commit git logs

**10. Ignoring Files**

https://docs.github.com/en/get-started/getting-started-with-git/ignoring-files

>> touch .gitignore

**10. BRANCHING**

Original branch is always **Master**

#Creates a new branch

>> git branch <new-branch-name>

#change branch to new branch

>> git checkout <branch-name>

#To create a new branch and change to it at the same time, you can use the **-b** flag:

>> git checkout -b <new-branch-name>

## go back to MASTER branch

>> git checkout master

T**o merge a different branch to the current branch**,

#normally done when a branched changes are final and needs to be merged to a current or main branch

#make sure you are in the current branch

>>git merge <branch-name> #name of the branch that needs to be merged to current branch

**Delete a branch**

>>git branch -d <branch-name>