

# Introduction to Git and Github

Some summary of my first introduction to Git and Github following the Net Ninja (<https://youtu.be/3RjQznt-8kE>)

Let's Git!

-- Here we learn

1. How to create Git repository
2. Git Stage and Commit
3. Understanding Git branching and branch conflict
4. Github repository, Git Push

Start by installing and setting up Git. <https://git-scm.com/>. Furthermore, you will need a text editor - for this, Visual Studio Code or Atom is fine. In particular, I used VS Code (<https://code.visualstudio.com/>) for this introduction. Here is also a link to Atom (<https://atom.io/>)

**Configure Git:**

```
git config --global user.name git_username  
git config --global user.email registered_git_email
```

If you'd like to check on your local machine which username and email is attached to Git, enter into the terminal: *git config user.name* and *git config user.email* respectively

## **Creating a Git Local Repository**

Create a directory: ***mkdir dir\_name***  
Inside the directory, create git repository : ***git init***

## **The 3-basic status repository**

First check Git status: ***git status***  
With that, we would know which file/directory has been either Modified, Staged, or Committed

Modified: This is creating and editing a file

Staging: ***git add file\_name*** for staging a file or ***git add .*** for staging all files  
To remove from staging area: ***git rm --cached file\_name***

Commit: ***git commit -m "Description of our file"***. Commit basically means save.

## **Navigating through different commit**

Checking change history: ***git log*** or for one line ***git log --oneline***

Navigate to a certain commit: ***git checkout commit\_id*** . Commit ID looks like e592bb3  
Navigate back to the HEAD: ***git checkout master***

Git revert: ***git revert commit\_id*** This revert any edit we made in that branch  
Git reset: ***git revert commit\_id***

## Branching

Creating branch: ***git branch branch\_name*** or ***git checkout -b branch\_name\_1*** to create different branches. Note that, this branch is created on the head (one can choose the relevant stem)

To see where we are on the created branch: ***git branch -a***

Switching to the relevant branch: ***git checkout branch\_name*** (master for the head)

Switching back to the head: ***git checkout branch\_name***

Deleting a branch: ***git branch -D branch\_name***

## Git Pushing:

Git Push: ***git push git\_url\_push-address master***

Instead of typing the long address and push command line all the time, we can as well use an alias. Here, the alias is **origin** which is the default on Git.

Git Push Alias: ***git remote add origin git\_url\_push-address***

Git Push Now: ***git push origin master***