# AP Operation

* Every month push into github of all repositories (ex C:\Poov\FldrGit\apgmail\AP\_Notes)
  + Go to the folder – Rt Click - Git Bash Here
  + git status
    - shows modified files in red
  + git add .
    - Files will be added into staging (local repository)
  + git status
    - Modified and updated in local repo will be in green color
  + git push
    - Push the updated files from staging to github.

## ???

* I committed one file n times in main, how can compare with earlier version?

## Installed in T480

### Git Bash

### GitHub Desktop

* Used the same existing repostirory AP\_Notes – mapped and pulled

## Git/Github

* Create Repository either in local or Pull from GitHub
* Add files / modify files in Local –
* Commit the changes in local to staging (local)
* Push from staging to GitHub

## a

# Ref URLs

<https://learngitbranching.js.org/> - about branching

<https://www.tutorialspoint.com/git/>

Git Desktop : https://desktop.github.com/ TortoiseGit: https://tortoisegit.org/

Git Cheat Sheet: https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf

Getting Started: https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Basics: https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository

Branching: https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell

Github Setup: https://git-scm.com/book/en/v2/GitHub-Account-Setup-and-Configuration

Git Tools: https://git-scm.com/book/en/v2/Git-Tools-Revision-Selection

Git Commands: https://git-scm.com/book/en/v2/Git-Commands-Setup-and-Config

a

# Books Read

## Distributed Version Control with Git\_ Mastering the Git command line - .pdf

# GitHub Repository

* GitHub is a web-based cloud service to host your source code(Git repositories). It is a centralized system.
* Git doesn’t require GitHub but GitHub requires Git.

## Create account in GitHub

## Create New Repository in Github

Create New Repo -> Repositories -> NEW

To access the code, we need Personal Access Tokens Settings -> Developer Settings -> Personal access tokens-> Generate new token

Copy the token since it cannot be viewed once we navigate the page

Windows Control Panel-> Credential Manager ->windows credential ->Add windows credential -> github.com -> Username -> pwd: Token.

# Git

* Version control tool (software) to track the changes in the source code.
* If you want to delete a Git (local) repository, you can simply delete the folder which contains the repository
* Git doesn’t require GitHub but GitHub requires Git.
* Initialize git - set identity (in cmd window)
  + If you use only one id use global - if you want to use multiple ids for multiple project remove global
* $ git config --global user.name "John Doe"
* $ git config --global user.email johndoe@example.com
* move to proper command place
  + git mv file\_from file\_to ???
  + git remote -v ???
  + git log --since='last month' --pretty=format:'%h;%an;%ad;%s' --author='Ionut Colceriu' > ~/log.csv

## Install Git (GitBash)

https://git-scm.com/download/win

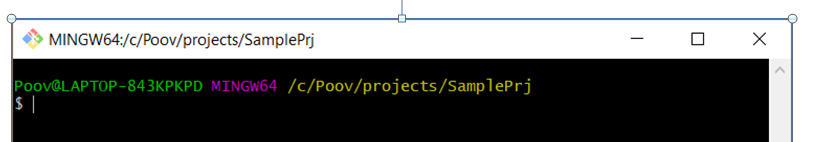
## Keys

q – exit from running comment – example after giving git log HEAD it will not come out – we need to give q

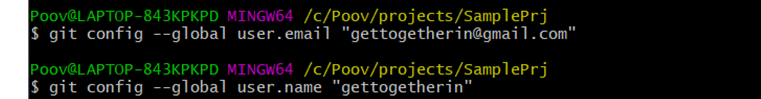
## git init

## git clone

Goto folder in windows explorer – Rt Click – Git Bash Here



Check global and local userid and email id already set. If not set that

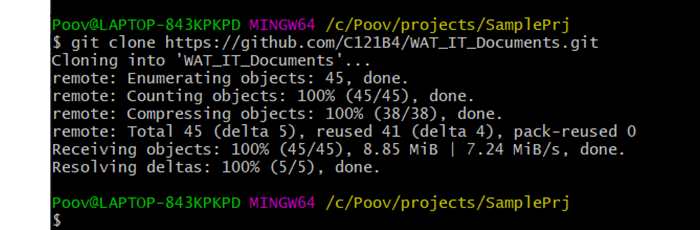


Clone the project by using github URL

git clone <https://github.com/C121B4/WAT_IT_Documents.git>



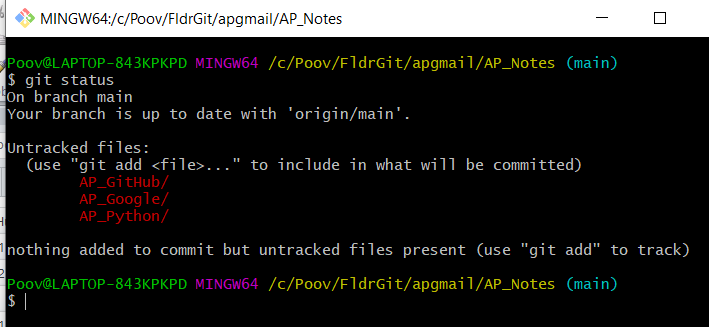
Asking to login in the website – once login your github account it will clone in your local repository



## git log

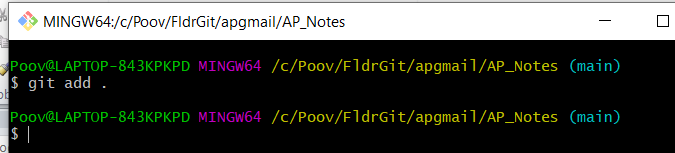
## git status

* to check for any untracked files

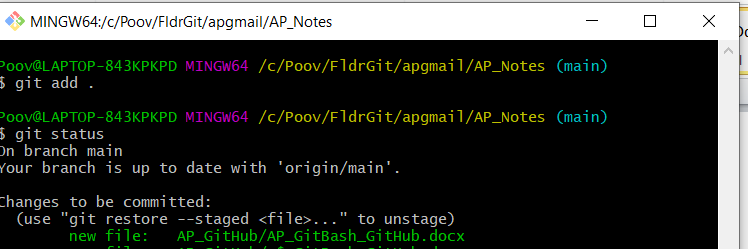


## git add .

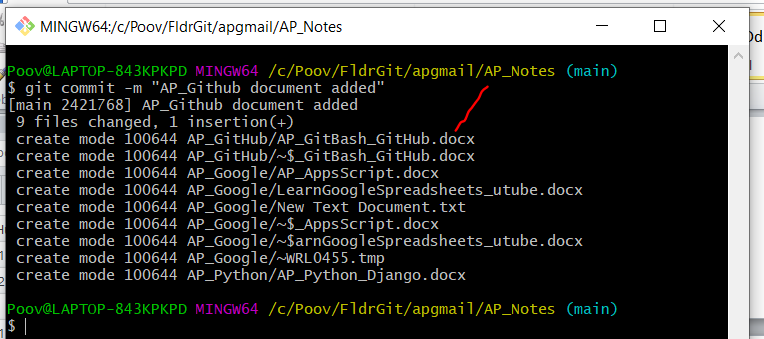
* adds the untracked files



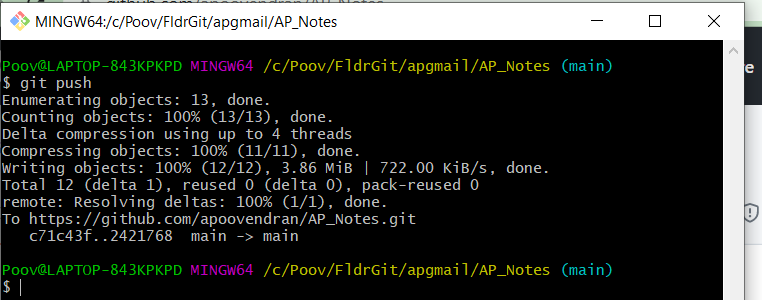
git status to know the status

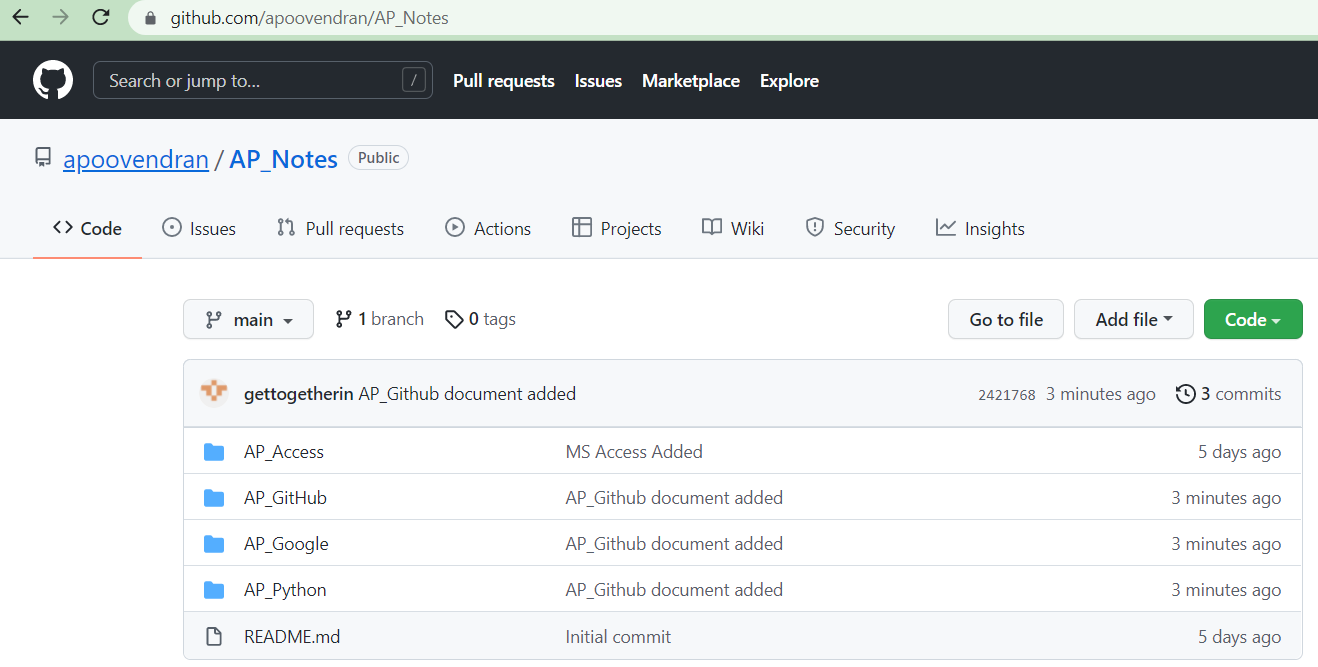


## git commit -m "message“

* creates a copy that needs to be updated in repository
* 

## git push

* pushes the code changes into the repository
* 
* In the github URL



## Handling with Branches

* In case of merge conflicts, need to resolve conflicts and then merge.
* For review purposes we usually create a pull request in github after pushing the code into the repository, review the code and then merge the branches

### git branch

### git checkout -b "newdevbranch"

* creates a new branch in local repository

### git pull

* gets the latest version code from GitHub

### To merge new branch to Master

#### git checkout master

* switches to existing branch

#### git merge newdevbranch