# Worksheet 1: SCM with Git

All software development projects need a mechanism to manage their codebase. Code versioning is critical in keeping track of progress. Source Code Management (SCM) is a system which keeps track of code. One of the most popular SCM is Git which was developed by Linus Torvalds, the creator of LINUX, to manage source code from other contributors. Git provides branching, multiuser, tagging, etc.

In our class, we will use Git as our main SCM tool.

## Git Command Summary

git clone  
*Clone a copy of a online repository and add a remote to it.*

git fetch  
*Fetch information of all branches, but not yet download the code.*

git pull   
*Download the latest code of the current branch. Pull before you start working.*

git branch  
*Show the current branch, change current branch*

git add <files>  
*Add files to the index*

git commit -m “<commit message>”  
*Save the current state to the local repository*

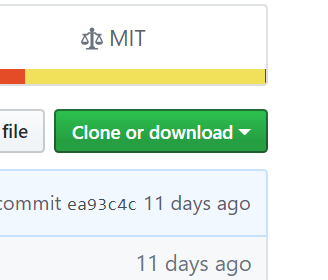
git push  
*Send the latest commit code to remote*

However, there can be conflicts during git push. Therefore, each contributor needs to create a Pull Request to notify other contributors to review his/her code before merging into the system. The Pull Request can also trigger other subsystem such as automated tests.

There are a lot of use cases of Git; we will explore that as we go through this course.

## Let’s do it

We are going to create a Git repository with an Cloud SCM provider (eg GitHub, BitBucket, etc). The repository will be used as your work submission for the entire semester. Highly recommended to maintain the nice and clean repository as it can be a great portfolio for your future career. CTOs nowadays check your repository before recruiting you.

1. Create an online repository
   1. Sign up with GitHub or BitBucket.
   2. Create a repository named “hybrid-mobile-dev”
2. Clone to your workstation
   1. Open a Command-Line Interface (CLI): run cmd
   2. Check whether git is available: just type git
   3. Back to the browser, get the URL:   
      Click “Clone or download” and copy the URL.
   4. Back to CLI,   
      Change to the drive you wish to save your work, for example D:  
      d:  
      Create a directory you wish to save your work,  
      mkdir CS4404  
      Change the current working directory to it  
      cd CS4404  
      Now clone the repository  
      git clone [https://github.com/<your\_username>/hybrid-mobile-dev](https://github.com/%3cyour_username%3e/hybrid-mobile-dev)  
      Git will start downloading the code
   5. Change the current working directory to your hybrid-mobile-dev.
   6. List the directory content  
      dir
3. Create a README.md  
   notepad README.md
4. Write some description about this repository. MD file is a mark-down file. It has simple syntax. See https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet.   
   # Chayapol’s Hybrid Mobile Development  
   This repository is a part of CS4404 class of 1/2017.  
   ….
5. Back to CLI, add README.md to Git index  
   git add README.md  
   or add everything  
   git add .
6. Commit (save locally). NB never commit with empty message! It’s rude!  
   git commit -m “Added README.md”  
   This can be done regularly. You should commit often.
7. Push! Not as often as commit. In a large project, a push will require Pull Request which other contributors in the team must spend time to review it.  
   git push

## Summary

1. You have learned basic CLI commands.
2. You have learned Git.
3. You have created your Git repository.