GIT/Version Control

Distributed Version Control System

# How to setup GIT

$ git init [project-name]

*Creates a new local repository with the specified name: C:\Desktop\> git init project-name would create it on the desktop*

$ git clone [url]

Downloads a project from said URL

$ cd project-name

*Change Directory into the newly created directory*

# GITHUB

## Configure

Configure user information for all the local repositories

$ git config --global user.name dkentuk

Sets the name you want atached to your commit transactions

$ git config --global user.email [dk654@hotmail.co.uk](mailto:dk654@hotmail.co.uk)

Sets up the git with correct email

## Make Changes

Review edits ready for commit

$ git status

Lists all new or modified files to be commited

$ git add [file] .=all file.txt =single file etc

Takes a snapshot of the file, ready to be commited

$ git commit -m "[descriptive message]"

Commits file to the local repository, ready to be pushed to server

$ git diff

Shows file differences not yet staged

## Deletion/Refactoring of Files (Useful for me)

Relocate and remove versioned files

$ git rm [file]

Deletes the file from the working directory and stages the deletion from the GITHUB server

$ git rm --cached [file]

Removes the file from version control but preserves the file locally. Useful when you accidentally upload your entire desktop to GITHUB

$ git mv [file-original] [file-renamed]

Changes the file name and prepares it for commit

## Synchronize Changes

Register a repository bookmark and exchange version history

$ git pull [url]

Downloads the information and history off of the server url

$ git push [alias] [branch]

Uploads all local branch commits to GitHub

### More Indpeth – If you want to look at more than just basic functionality

https://github.com/tiimgreen/github-cheat-sheet