**How Git Works**

A Git repository is a database containing all the information needed to retain and manage the revisions and history of a project. In Git, a repository retains a complete copy of the entire project throughout its lifetime. However, unlike most other version control systems, the Git repository provides not only a complete working copy of all the files in the repository but also a copy of the repository itself with which to work.

**Things You Should Know**

* A repository is a container for  a project you want to track with Git.
* When you create a repository in a folder Git sees that folder and all sub-folders as the scope of the project.
* Git allows you to have as many repositories for as many projects as you want – each will be tracked /treated independently.
* Repositories are often referred to as Repo’s.
* A repository can be stored locally or on a server (typically an online service)
* Commits are what we call and do when we are saving our work to the Repo
* Commits should be done at logical points – perform a COMMIT when a piece of a project has been completed
* Commits become a point in the Repo that we can go back to
* There are three stages in which we progress code changes to the Repo
  + Modified Stage – the point at which a file has been modified and saved
  + Staging Stage – the action of letting Git know the file is ready to be Committed to the Repo
  + Commit Stage – the action of telling Git to Commit the changes to the Repo
* The logical line/path on which our Commits are saved on is called a branch
* Branches allow us to work on features without interrupting the main branch where our production/primary code is maintained
* There can be more than one branch.
* There is one main branch that is called the master branch.