**Part 1:**

Create a GitHub account : Done

**Part 2:**

Install Git bash <http://git-scm.com/downloads> and browse the documentation : Done

**Part 3:**

**What is Github?**

GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project.

The flagship functionality of GitHub is “forking” – copying a repository from one user’s account to another. This enables you to take a project that you don’t have write access to and modify it under your own account. If you make changes you’d like to share, you can send a notification called a “pull request” to the original owner. That user can then, with a click of a button, merge the changes found in your repo with the original repo.

These three features – fork, pull request and merge – are what make GitHub so powerful

**When was it created?**

GitHub was developed by Chris Wanstrath, PJ Hyett, Tom Preston-Werner and Scott Chacon using Ruby on Rails, and started in February **2008**. The company, GitHub, Inc., has existed since 2007 and is located in San Francisco.

**Why?**

Before the idea of open source software flipped the tech world upside-down. [The open source Linux operating system](https://www.wired.com/2012/03/mr-linux/all/) was already running an enormous number of machines on Wall Street and beyond, proving you can generate big value—and big money—by freely sharing software code with the world at large. But the open source community was still relatively small. When coders started new open source projects, they typically did so on a [rather geeky and sometimes unreliable internet site called SourceForge](http://en.wikipedia.org/wiki/SourceForge).

DiBona, [an open source guru inside Google](https://www.wired.com/science/discoveries/news/2007/03/73007), was worried that all of the world's open source software would end up in that one basket. "There was only one, and that was SourceForge," he says.

So, like many other companies, Google created its own site where people could host open source projects. It was called Google Code. The company had built its online empire on top of Linux and other open source software, and in providing an alternative to SourceForce, it was trying to ensure open source would continue to evolve, trying to spread this religion across the net.

But then [GitHub came along](https://www.wired.com/2012/02/github-2/) and spread it faster.

**By who?**

GitHub was developed by Chris Wanstrath, PJ Hyett, Tom Preston-Werner and Scott Chacon using Ruby on Rails

**What similar platforms exist?**

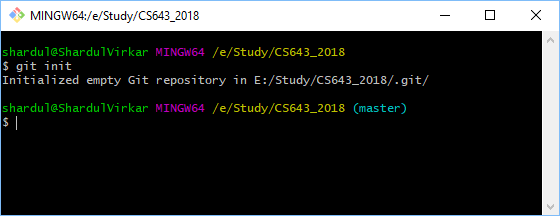
Bitbucket, SourceForge, GitLab, Kiln, Codeplane

**Why would you use such a platform?**

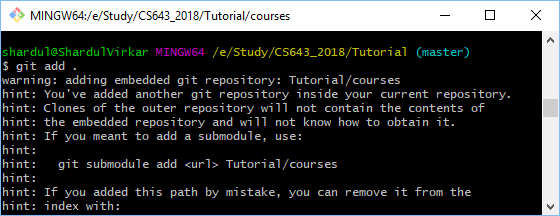
You can actually use GitHub for any types of files. If you have a team that is constantly making changes to a word document, for example,  you could use GitHub as your version control system. This practice isn’t common, since there are better alternatives in most cases, but it’s something to keep in mind.

**Part 4: Git Tutorial**

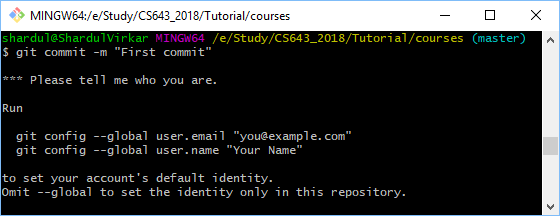
1. git init



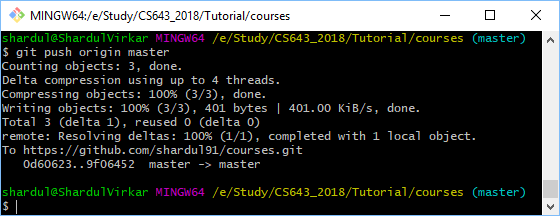
1. git add



1. git remote add origin



git push origin master



**Part 5:**

Repository: Repository is a data structure used by VCS to store metadata for set of files and/or directories. it stores the set of file as well as history of changes made to those file.

Commit: Basically git commit "records changes to the repository" while git push "updates remote refs along with associated objects". So the first one is used in connection with your local repository, while the latter one is used to interact with a remote repository.

Push: The git push command is used to upload local repository content to a remote repository. Pushing is how you transfer commits from your local repository to a remote repo. It's the counterpart to git fetch, but whereas fetching imports commits to local branches, pushing exports commits to remote branches.

Branch: Branching means you diverge from the main line of development and continue to do work without messing with that main line. The way Git branches is incredibly lightweight, making branching operations nearly instantaneous, and switching back and forth between branches generally just as fast.

Fork: Forking the official repository creates a copy of it on the server. This new copy serves as their personal public repository—no other developers are allowed to push to it, but they can pull changes from it (we’ll see why this is important in a moment)

Merge: Join two or more development histories together

Clone: Clone a repository into a new directory

Pull: Fetch from and integrate with another repository or a local branch

Pull request: Generates a summary of pending changes

**Part 7:**

$ git init

$ git add .

$ git commit -m ‘Initial commit’

$ git remote add origin <https://github.com/shardul91/CS6432018.git>

$ git push origin master

References: <https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>

<https://en.wikipedia.org/wiki/GitHub>

<https://www.wired.com/2015/03/github-conquered-google-microsoft-everyone-else/>

<https://beebom.com/github-alternatives/>