Andrew Preciado

Pace University

Git and Github

**Q: What is Github?**

Github is a code sharing and publishing service for developers. It also allows developers to take up a project as a team.

**Q: When was it created? Why? By who?**

Github launched in 2008 by Tom Preston-Werner, Chris Wanstratch, and PJ Hyatt. Github was made for coders so they could host and manage their software projects using Git.

**Q: What similar platforms exist? Why would you use such a platform?**

Bitbuck, Gitlab, SourceForge, and Launchpad are some similar platforms to Github. There are differences in private vs public repositories, security, flexibility with projects, and different features such as issue tracking.

**Terms**

Repository

-used for organizing a single project. They can contain folders, files, images, and anything you project needs.

Commit

-Also known as saved changes

Push

-Referencing sending your commited changes to a remote repository, which is a verison of your project that is hosted on the Internet or network somewhere.

Branch

-A way to work on different versions of a repository at one time

Fork

-A copy of a repository. Allows you to freely experiment with changes without affecting the original project.

Merge

-Push the changes to your Branch in a repository.

Clone

-Creates a local copy of your repository on your computer and allows you to sync between the two locations.

Pull

-Allows you to tell others about changes you have pushed to a repository.

Pull Request

-Propose your changes and request that someone review and pull in your contribution and merge them into their branch.

**PART 4 TUTORIAL NOTES**

git init

-to initialize a Git repository (already made)

-creates .git folder in the repository (guts of Git, where everything happens)

git status

-see current status of project

-sees .txt and other files in the repository

-good to run often

git add name.txt

-adds file to the Staging Area (place to group files together before you commit them to Git, meaning they not yet in the repository)

-you can add or remove files from the stage before we store them in the repository

-Git is now tracking the file.

-do a git status after adding a file to see where you at

git commit –m ‘message’

-after commit you can use wildcards to add many files of the same type.

Git add ‘\*.txt’

-will add files all with .txt at the end

-after the add, do a git status to see what you will be committing

git log (can use git log –summary, to see more info for each commit.)

-journal that remembers all changes we have committed so far in the order we commited them.

-shows the message as you committed each.

Git remote add (url)

-to push our local repo to a Github server we will need to add a remote repoistory

-typically name the main one as ‘origin’

git push –u origin master

-where to put commits when we are ready.

-name of remote is origin

-default local branch name is ‘master’

-the –u tells git to remember paramters so the next time we can just run ‘git push’ and git will know what to do

**Update Readme.md**

-<https://github.com/paceuniversity/courses>

-Click README.md

-Click the Pencil icon (says, “Edit this file in the fork of your project” if you hover over it)

-Add you Name, date, and time to the list.

-Click Propose file change button

-Click Pull Request

-Wait for the owner to merge the request