Version Control

Understanding how to use Git/Github

TAMID GROUP

Before We Begin: Basic Definitions

Git - a distributed version-control system for tracking changes in source code during software development

Repository - a directory or storage space where your projects can live

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.

Pull - the git pull command is used to fetch and download content from a remote repository and immediately update the local repository to match that content. Merging remote upstream changes into your local repository is a common task in Git-based collaboration work flows.

Merge - merging is Git's way of putting a forked history back together again. The git merge command lets you take the independent lines of development created by git branch and integrate them into a single branch.

Rebase - rebasing is the process of moving or combining a sequence of commits to a new base commit

Pull Request - you make local code changes and then submit those changes to a remote project maintainer for review before those changes are implemented, or merged.

What is Git?

- **Git** is a distributed version control system for tracking changes in source code during software development
- It is used for collaboration and to have a changelog of what files have been updated
- Typically used with hosts such as Github or Bitbucket to actually host files in ‘repositories’

Why is Version Control important?

FEATURE BRANCH WORKFLOW

Feature Branches provide an isolated environment for you to make changes to the codebase. You can test something out, without worrying about accidentally ruining any old code!

PULL REQUESTS

Pull Requests allow for developers to merge their feature branch into the ‘master’ branch. This is often when developers ask for reviews from co-workers/peers, and then update their code accordingly.

DISTRIBUTED DEVELOPMENT

Distributed Development means that each developer gets a local repository copy on their machine, with a full history of commits.

COMMUNITY

Git has come to be the expected version control system for new projects. In addition, Git is very popular among open source projects. This means it’s easy to leverage 3rd-party libraries and encourage others to fork your own open source code.

Live Example: Getting started with Git

1. Ensure Git is on your machine by running `git --version` in the terminal, and set your credentials using `git config --global user.name "[name]"` and `git config --global user.email "[email address]"`

2. Create a Github account on www.github.com
   a. It is recommended that you also get the Github Student Education Pack for a bunch of cool tools and free Github Pro!

3. Create a repository on Github by selecting the ‘+’ icon on the top right

4. Clone the repository using `git clone https://www.github.com/<profile-name>/<repository-name>.git`

5. Add a new remote, so we can access the remote repository `git remote add upstream https://www.github.com/<profile-name>/<repository-name>.git`

6. Add a file to this new folder, and run `git status` to see what changes have been made

7. Add this file using `git add .` (for all files) or `git add <file-1>...<file-n>`

8. Write a commit message `git commit -m “This is my first commit!”`

9. Add that to your remote repository by pushing it using `git push`

Live Example: Working with Collaborators

1. Pair up with a friend, and add them as a collaborator to your branch by going to “Settings” and “Manage Access”

2. The person who was added should repeat the step of cloning the repository, by using `git clone https://www.github.com/<owner-profile-name>/<repository-name>.git` and `git remote add upstream https://www.github.com/<profile-name>/<repository-name>.git`

3. Create a new branch using `git checkout -b new-branch`

4. Repeat the steps of adding a file, creating a commit message, and pushing it

5. Switch branches using `git checkout master` and go back to the original branch. You can check your current branch by running `git branch`

6. Combine the master branch with the new branch, by running `git merge origin/new-branch`.

7. Both users should run `git pull` and then `git log` to see that the new commit was added!

Extra Resources

[Git Tutorial for Beginners]

[Interactive Git Tutorials]

[Git Cheatsheet]

TAMID GROUP