Git Part 2

Intro to GitHub and Project Creation

November 7, 2019
Honolulu Women in Tech

Outline

- Review Git Intro
- GitHub
  - Create an account
  - GitHub overview
- Git Exercises
  - Three ways to create a Git project
  - Concepts of cloning, pulling, and pushing
- More GitHub Fun

Review Git Intro

Git

Keeping track of work

- Centralized storage for projects
- Version Control
- Collaborative Workflow

A few basic scripts (how to communicate with computers)

```
>pwd

>ls

>mkdir

>cd

>clear
```

A few basic scripts (how to communicate with computers)

`>pwd` -- Print your working (current) directory

`>ls` -- List directory contents

`>mkdir` -- Make a new directory

`>cd` -- Change directories

`>clear` -- Clear the terminal

A few basic Git commands

```
>git init

>git status

>git add /path/to/file.txt

>git commit -m “Write a descriptive message regarding what you changed”

>git log
```

A few basic Git commands

`>git init` -- Initialize a Git repository

`>git status` -- Show the status of files

`>git add` -- Add file contents to staging

`>git commit` -- Record repository revisions

`>git log` -- Show the revision history

GitHub

A centralized cloud-based storage to **store** and **share** your Git repositories

A brief aside: What is the cloud?

- Imagine that there were a million or more copies of your laptop or desktop all working away inside of one or more warehouse-sized buildings — a data center
- But to us as the end user, we don't have to manage all of the physical devices, networking (getting data from one device to another), computing power, storage capacity, memory requirements, and electrical power ourselves
- The cloud is simply a service that gives us access to the computer resources in a data center without having to worry about managing them ourselves
- That service, to us, comes in the form of a website: We just log in to a site like Google Cloud or Amazon Web Services and spin up computers that we need

Let’s all create an account on GitHub

GitHub: Different account types

- GitHub has Personal, Organization, and Enterprise account types, but today we'll focus on Personal
  - A Personal account is, for example, your own GitHub account at https://github.com/username
  - An Organization account has extra team-based features; for example: https://github.com/honoluluwomenintech

GitHub: Introduction

- Overview: View your code contributions and showcase your favorites out of the repositories that you own
- Repositories / “repos”: View a list of repositories that you own
- Stars: View a list of repositories that you’ve bookmarked
- Followers: View GitHub users who are following you
- Following: View the GitHub users that you’re following
- Profile Settings: Here you can edit your public profile, change your username (has side effects), change your password, set up notifications, block users, and access security settings

GitHub: Repositories

- A repository is the main hub of your current project
- Code: View the contents of README.md, different branches, and the last change made to each file
- Issues: Also known as “tickets”, this gives you and your team or the public an idea of who is working on what
- Pull requests: Also known as “PRs”, this allows others to easily review a bunch of your changes and vice versa
- Settings: Here you can add your project collaborators (no more than 3 others if your repository is private)

Git Exercises

Three ways to create a Git project

a. Create a new repository on GitHub and `git clone` it to have a local copy

b. Fork an interesting repository (must be owned by another person or organization) by clicking “Fork”, and `git clone` it to have a local copy

c. Import your existing project folder: Create a new repository on GitHub, create a local copy of the repo if you haven’t, run `git remote add`, and `git push` your local copy to the new remote

Project Creation: Create a new repository

1. Click on the + dropdown in the top right-hand corner of GitHub and click on “New repository”
2. By default, you’re the owner, and give your repository a name
3. While optional, having a description looks really nice
4. Set your repository to public or private (can be changed later)
5. It’s always good practice to add a README, to set a .gitignore based on the language you’ll use, and a license (usually MIT)
6. You’re all set to click “Create repository”
7. Now on the repository page, click on “Clone or download” and the clipboard icon, then in your terminal: `git clone repo-link`

Project Creation: Fork an interesting repository

As you https://github.com/explore GitHub projects or come across an interesting repository by some other means, if you want to start working with the code and saving any changes that you'll make, it's really easy to fork — meaning, to create your own remote copy of another's repository.

1. On the repository page, click "Fork" in the top right-hand corner; note:
    - a) You cannot fork a repository that you own
    - b) You cannot have already forked the said repository
2. Now on the repository page of your forked copy, click on "Clone or download" and the clipboard icon, then in your terminal: `git clone your-forked-repo-link`

Project Creation: Import your existing project folder

1. Create a new repository on GitHub (follow the same steps as two slides prior; however, let's skip step 5)
2. Go to your local (= laptop) project folder and run `git init` on it; this creates a local copy of the repository
3. Run `git add .` to keep track of all your files
4. Run `git commit -m "Message"` to save the files to Git
5. Run `git remote add origin https://github.com/username/repo_name.git` to assign the (commonly used) nickname origin to the GitHub address specified
6. Run `git push -u origin master` to upload your local copy of the repository to the new remote (= GitHub) address

Project Creation: Import your existing project folder

```
>cd /path/to/your/existing/project/folder

# Run only if your folder is not already a Git folder
>git init
>git add .
>git commit -m “Initial setup of my first project”

>git remote add origin
https://github.com/username/repo_name.git
>git push -u origin master
# View the remotes addresses that exist
>git remote -v
```

Concepts of cloning, pulling, and pushing

- To clone via `git clone` means to create a local copy of a remote GitHub repository

- To pull via `git pull` means to download repository changes from a remote copy to your local copy

- To push via `git push` means to upload repository changes from your local copy to a remote copy

More GitHub Fun

- Update your public profile
- Explore repositories that might be of interest to you
  - Star (bookmark) those repositories as you find them
- Follow your friends or other GitHub users
- We haven't covered some of these topics in depth but feel free to follow GitHub's own Hello World (getting started) guide: https://guides.github.com/activities/hello-world/