# Introduction to GitHub

## The Plan

1. Source Control, Git, and GitHub
2. The Basics of Markdown
3. Getting Started with GitHub

- Creating an account
- Creating a new repository from scratch
  - Branches, Commits, Issues and Pull Requests
- Working with a forked repository

## Source Control, Git and GitHub

**Source Control**, the management of changes to documents, computer programs, large web sites, and other collections of information. Examples of source control tools: Subversion, Perforce, Git...

**Git**, http://git-scm.com/, is a version control system that tracks changes to files in a project over time. Git is a command line tool.

**GitHub**, http://github.com, is a web-based Git repository hosting service with all of the revision control and source code management of Git plus some added features. GitHub provides a web-based graphical interface and desktop as well as mobile integration. See GitHub repositories in action at https://github.com/HeardLibrary

## The Basics of Markdown

Markdown, http://daringfireball.net/projects/markdown/, is a lightweight and easy-to-use syntax for styling all forms of writing on the GitHub platform.

You can use Markdown for:

- Gists - a quick and easy way to share snippets with collaborators
- Comments in Issues and Pull Requests
- Editing the repository Wiki
- Files with a `.md` or `.markdown` extension such as the repository `Readme.md` file

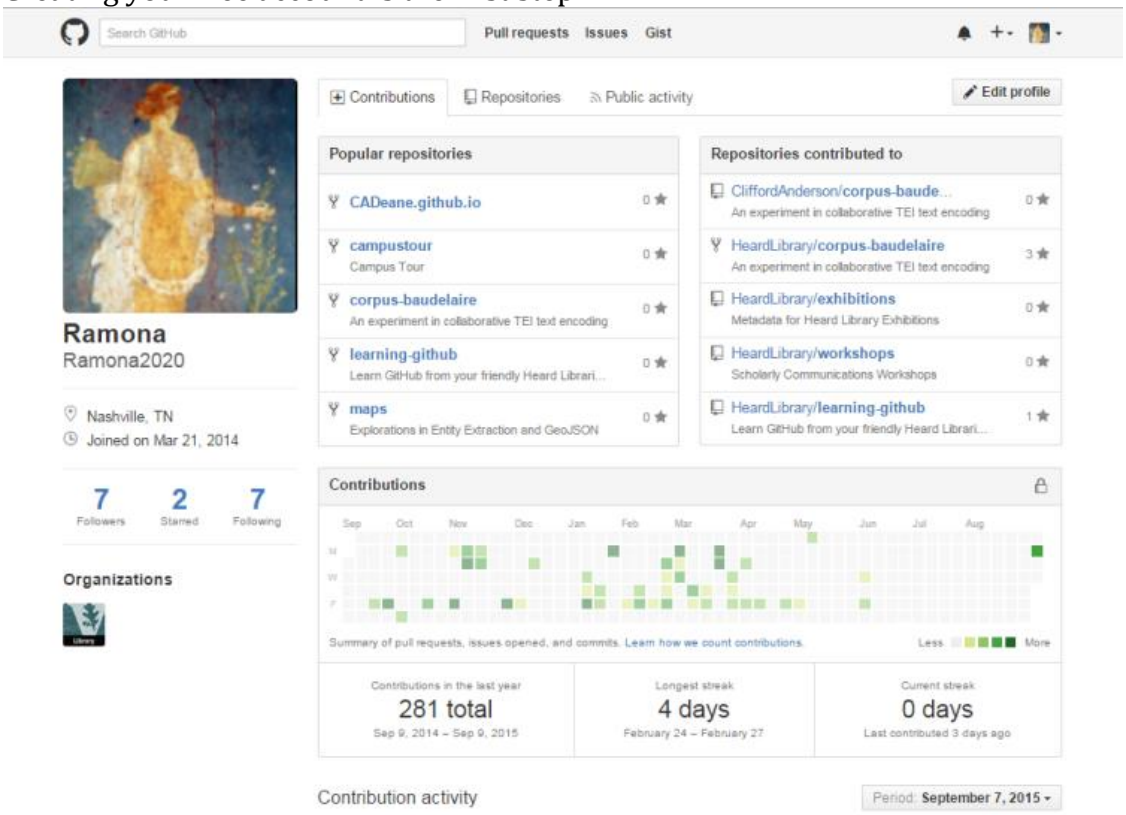
GitHub.com uses its own version of the Markdown syntax that provides an additional set of useful features, many of which make it easier to work with content on GitHub.com. Visit the Mastering Markdown guide: https://guides.github.com/features/mastering-markdown/

## Getting Started with GitHub

GitHub is free to use for public projects. A free account give you the ability to create an unlimited number of public repositories with an unlimited number of collaborators. GitHub

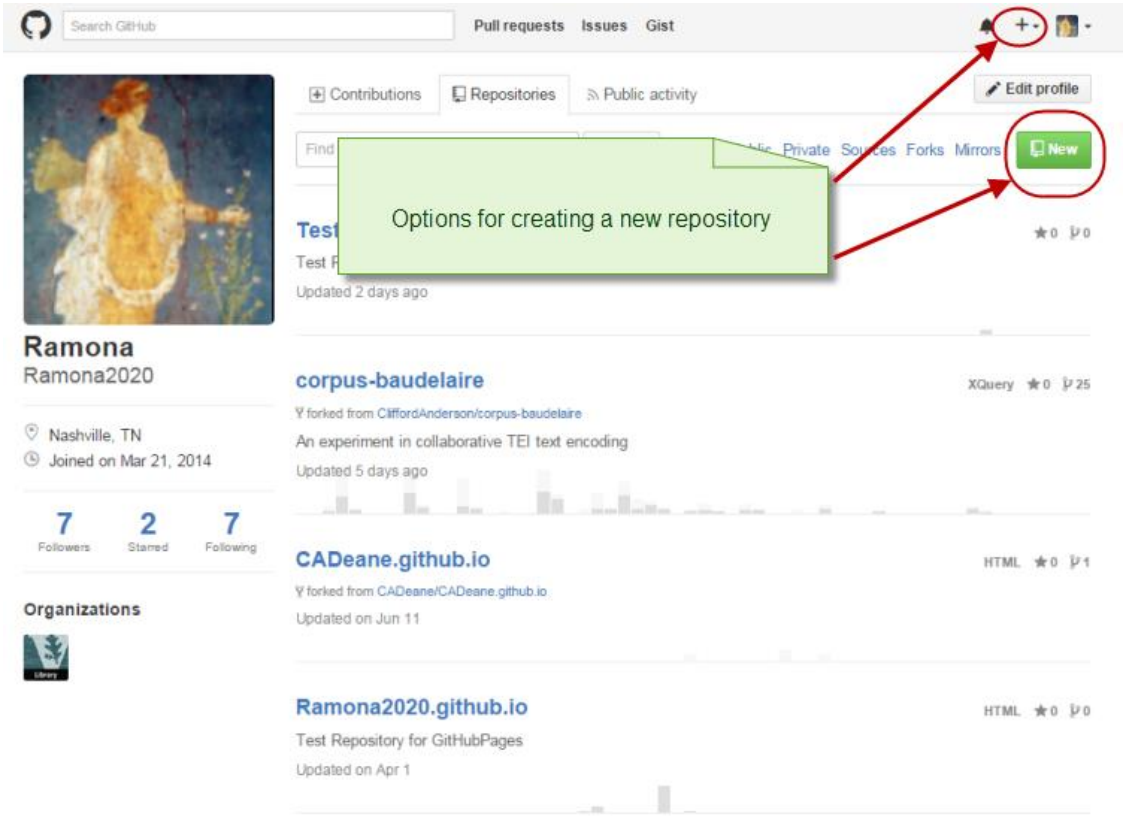
also offers a number of personal and organizational plans for those who need to collaborate on private repositories.

Creating your free account is the first step



## Creating a New Repository

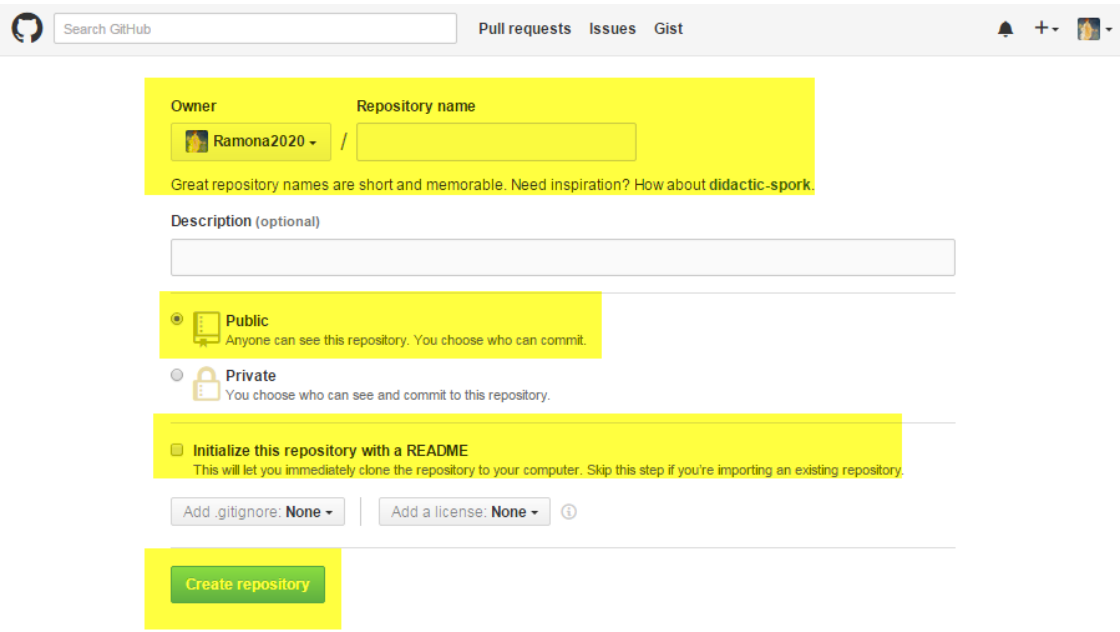
You have two options when creating a new repository. You can click on the + next to your profile image in the top right corner of the screen or go to the repositories tab within your profile page and click the New button.




*Creating a new repository in GitHub*

Just a few easy steps to initialize your repository:

1. Name it - Always keep it short and sweet.
   - Let's create a repository named **Hello-World**
2. Describe it (optional)
3. Select **Public** for your repository status
4. Check the box to initialize your repository with a README
5. Click Create Repository!




*New Repository Form*

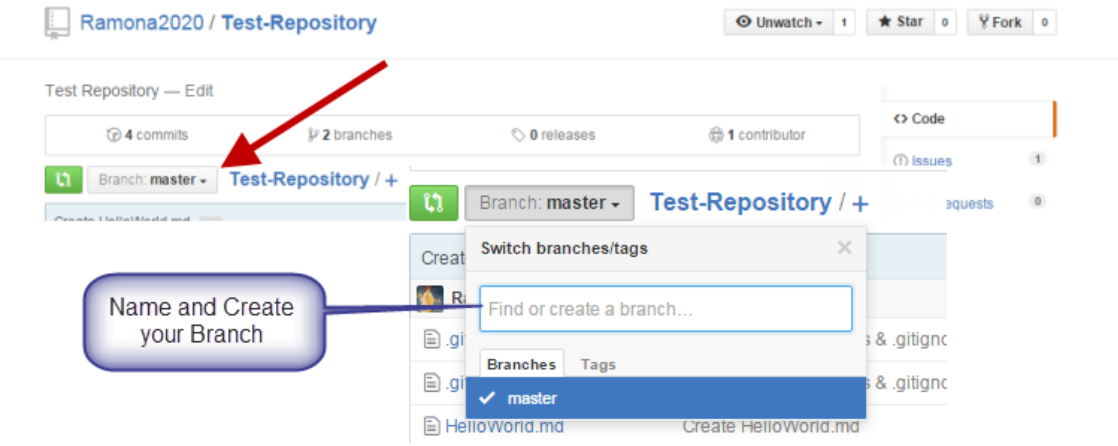
## Let's Play with Your New Repository

- Let's create an **Issue**. An Issue is a note on a repository about something that needs attention. The Issue could relate to a bug, a feature request, and questions.
- Open an Issue for README edits by clicking on the Issues link in the column on the right


*Imgur*

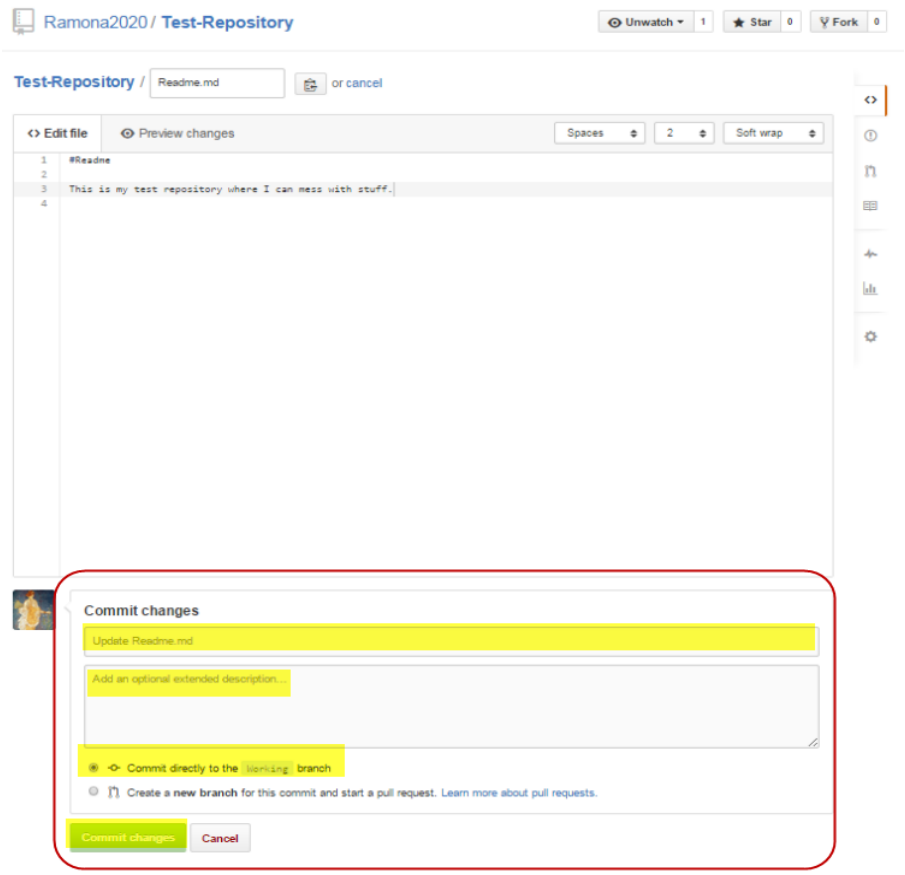
- **Branching** is the way to work on different parts of a repository at one time. By default all new repositories are created with one branch named `master`. You can work within your master branch, but creating working branches allows you to develop a feature or idea and work out all the bugs before merging it into `master`(your production branch).

- Go to your new repository `hello-world` and click the drop down at the top of the file list that says **branch: master**
- Create a branch named `readme-edits`



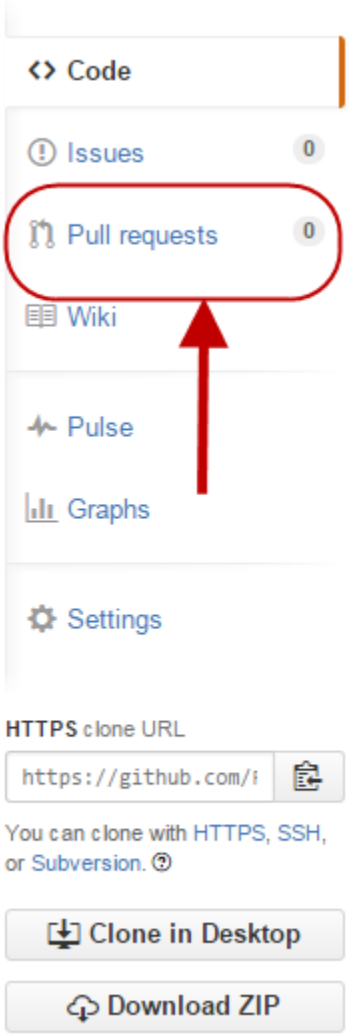

*Imgur*

- On GitHub, saved changes are called **commits**. Each commit has an associated **commit message** that describes the change being made and why. These commits document the history of the repository.
- We are currently in our working branch `readme-edits`, let's edit our README file by clicking on the file and then clicking on the pencil icon in the upper right corner of the file view to edit. Make a change to the file and commit it!



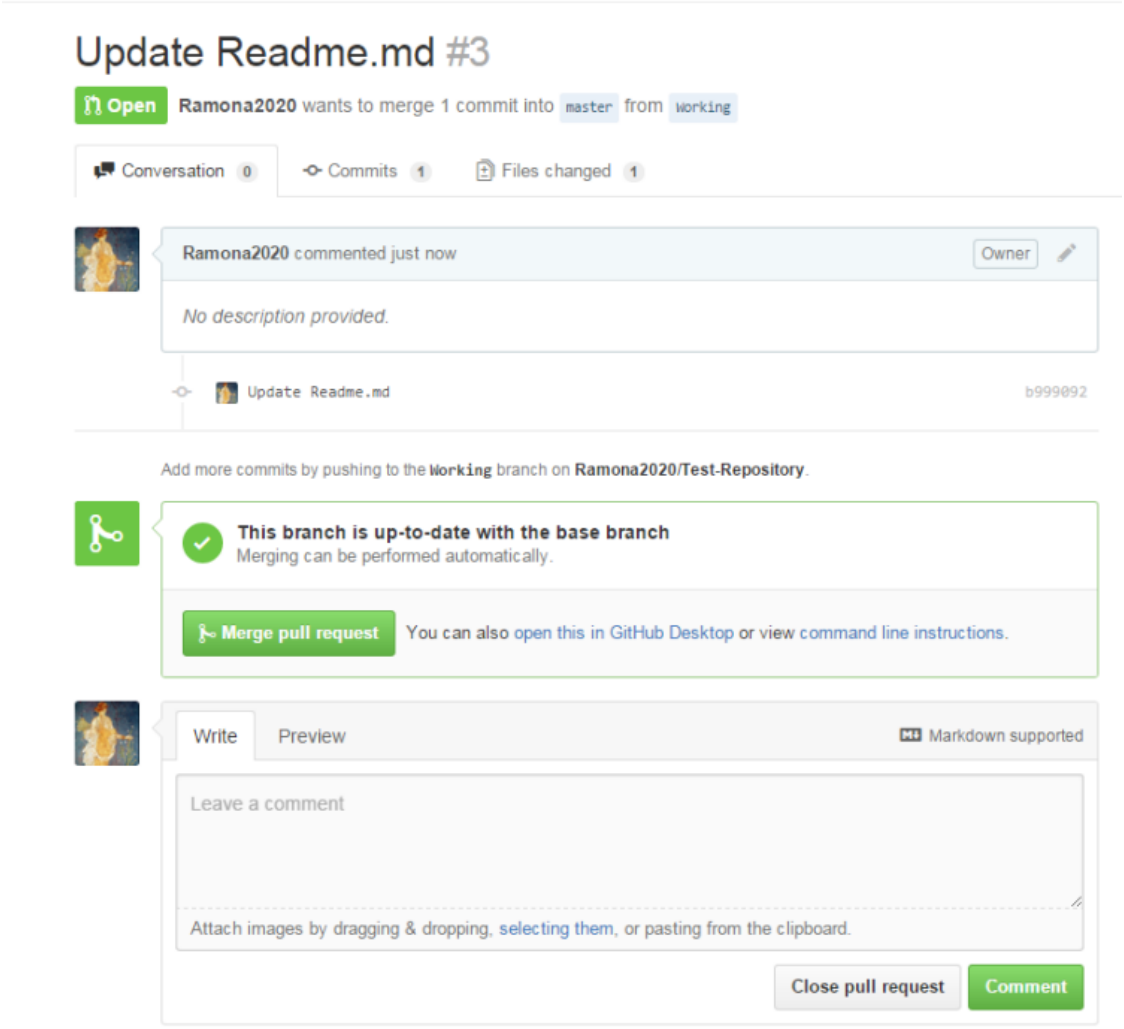

*Imgur*

- **Pull Requests** are the key to collaboration on GitHub. Using a **pull request**, you are proposing changes and requesting that someone pull in your contributions. In GitHub, pull requests allow you to compare the content of two branches, and the changes, additions and subtractions are shown in greena nd red and called **diffs**.
- Think of pull requests as a way to start a conversation where you can get feedback on the development process. You can also use GitHub's @mention system to ask for feedback from specific people. *We're going to open a pull request for our own repository and merge it ourself for practice. Click on the Pull Request icon in the side bar on the right.


Code
Issues
0
Pull requests
0
Wiki
Pulse
Graphs
Settings
HTTPS clone URL
https://github.com/
You can clone with HTTPS, SSH, or Subversion.
Clone in Desktop
Download ZIP

*Imgur*

- The last step is to put it all together and merge your `readme-edits` branch in the `master` branch. Once your merge is complete you can delete your working branch.



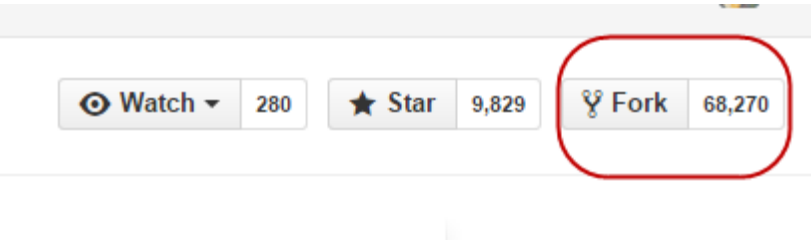

*Imgur*

## Working with a Forked Repository

Building your own repository can be fun, but sometimes you just want to contribute to an existing project or maybe you'd like to use someone else's project as the starting point for a new project of your own. If that's the case it's time to Fork.

Creating a **fork** creates a personal copy of someone else's project. You can submit Pull Requests to help make other people's projects better by offering your changes up to the original project. We're going to create a fork of the **Spoon-Knife** repository.

- Navigate to the page for the **Spoon-Knife** repository, and click the **Fork** button in the header of the repository.



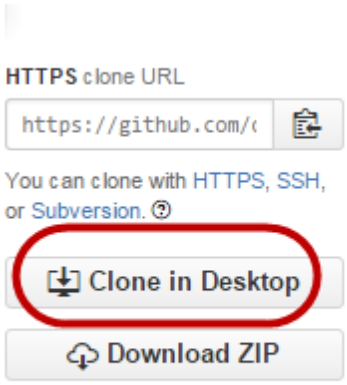

*Imgur*

We've been able to make edits to our files in the web version of GitHub, but that can get tricky. Installing the GitHub desktop client gives us more flexibility in how we can interact with the documents and files within our repository.

Download GitHub Desktop (available for MAC and PC) https://desktop.github.com/

Now that you've successfully installed GitHub Desktop and forked the Spoon-Knife repository, it's time to get to work. Currently your fork of the Spoon-Knife repository only exists on GitHub, but we need to clone it to your computer.

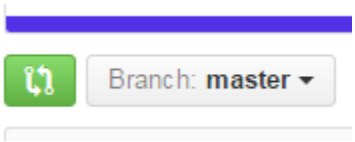
- Navigate to your fork of the Spoon-Knife repository. Click on **Clone in Desktop** in the right hand side bar.




*Imgur*

- Once you've successfully cloned the repository, all the repository files will be available to you on your desktop. Let's open a text editor (we'll use Notepad today, but I recommend installing ATOM, https://atom.io/, for your personal use) and make some changes to the *index.html* file.
- Remember to create a working branch first!
- Commit your changes and then click on the **Sync** button to push your changes from the desktop client to GitHub.com.
- Now it's time to propose changes into the main project.

- Go to your fork of the repository on GitHub.com and click on the **Compare and Pull Request** button




*Imgur*

Once you've submitted the pull request it is up to the project owner whether your changes will be pulled into the repository or not.

Success! You are now a GitHub user!

For more helpful tutorials about using GitHub visit **GitHub Guides**
https://guides.github.com/