# GitHub.com

## See nycda.com/git for a step-by-step guide

# Learning Objective

- Setup our GitHub profiles
- Make our first push to our first repository

# Why learn GitHub?

- GitHub has become the global destination for developer portfolios
- It allows you to easily collaborate with other developers on projects and for many companies is the standard for this kind of collaboration
- Using it is a great way to learn software development process fundamentals and best practices

# What is GitHub?

- First of all, Github is **not** `git` (remember that `git` is a command-line source control program that lives on your computer)
- GitHub is a **website that hosts `git` repositories**
- It allows you to easily push `git` repositories to it and then view the commits and code contained in an easy to use interface
- The platform also comes with many project management tools like issues, pull requests, and wikis

# Open source + GitHub

- 'Open source' projects are often hosted and managed on GitHub
- These projects are typically non-commercial and encourage contributions, or "pull requests", from anyone around the world
- Contributing to an open source project is a great way to give back to the community while also building credibility

# Getting ready to use GitHub

- First, you'll need a `git` repository locally to push to GitHub.com
- Assuming this repository is created, you'll need to create a new repository on GitHub.com
- The repository on GitHub.com will basically be an empty shell, allowing you to push your local repository into it

# Creating a repository on GitHub

- Go to github.com, sign in, and click the + at the top right of the page, select "new repository"
- Give your repository a name, **make sure the option to add a README is unchecked**
- Once you get to your repository's new GitHub page, copy the "HTTP clone url". The URL should look kind of like this: `https://github.com/nycda/nycda-2015.git`

# Pushing to your new repository

## Follow along on your machine!

- Go to your repository using the Terminal and the `cd` command, verify you're in the right place using `git status` and `pwd`
- Make sure that you have at least one commit before you push
- To push your repository to GitHub, use the `git push` command with your HTTPS URL and the word 'master' after it:

```
git push https://github.com/<your username>/<your repo name>.git master
```

- Refresh the page on GitHub and you should see the file(s) you just pushed!

# What just happened?

- The `git push` command takes two arguments
  - The repository URL, known as the "remote"
  - The branch the repository is currently on - for now, always assume that this is "master", the default branch
- When the command is executed (you press enter), your local repository is pushed to and stored on GitHub.com

# Tip: Make pushing easier with remotes

- Rather than having to paste the HTTPS url every single time you'd like to push to a remote location, you can add a "shortcut" to it
- These "shortcuts" are known as **remotes**, this is how they're used:

```
git remote add <remote name of your choice> <remote url>
# for example
git remote add origin https://github.com/zachfeldman/testing.git
```

- `origin` is a common name to use for GitHub or the place your project's code is centrally stored

```
# now you can use this syntax instead!
git push origin master
```

# Common pitfalls

Your push could be rejected for a number of reasons, including:

- You initialized your repository with a README.md file which GitHub won't allow you to overwrite, try creating a new repository on GitHub.com from scratch
- Your code is behind GitHub's version of your code (only likely if multiple people are working on a project or you're using multiple computers)
- You are pushing a repository with no commits (why would you do that?)

You can force a push, but **be very careful with this** as you'll overwrite everything that lives on GitHub.com

```
git push origin master -f
```

# The README.md file in your repository

- The `README` file has traditionally been used in software development to provide information on a project beyond file and folder names
- On GitHub, it's used as a "project homepage" to describe a project and possibly how to get a version of it up and running using the files provided
- This file is formatted with Markdown, an HTML-like markup language
- For more information on Markdown, see GitHub's Guide

# Resources

## TeamTreeHouse

Git Basics - Working with Remote Repositories

## Other Resources

Github for Developers

# Quiz

1. What's the difference between `git` and `github`
2. How do you push to GitHub, assuming you've named the GitHub URL, "origin"?