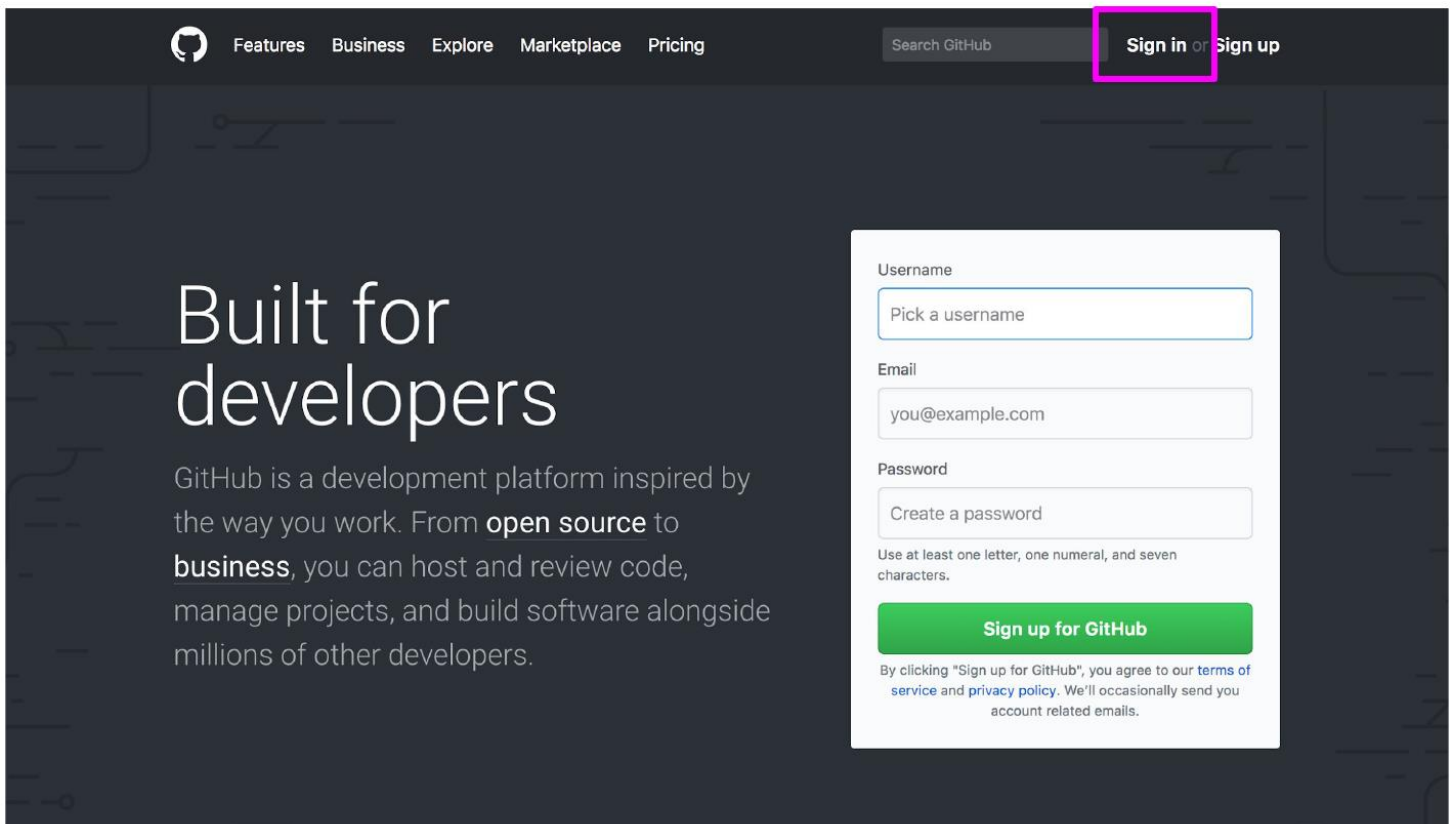# GitHub and Git

Now that we've got a handle on what version control is, in this lesson, you will sign-up for a GitHub account, navigate around the GitHub website to become familiar with some of its features, and install and configure Git; all in preparation for linking both with your RStudio!

What is GitHub?

As we previously learned, GitHub is a cloud-based management system for your version controlled files. Like DropBox, your files are both locally on your computer *and* hosted online and easily accessible. Its interface allows you to manage version control and provides users with a web-based interface for creating projects, sharing them, updating code, etc.

Signing up for GitHub

To get a GitHub account, first go to https://github.com/. You will be brought to their homepage, where you should fill in your information - make a username, put in your email, choose a secure password, and click "Sign up for GitHub."

github.com

Signing up for GitHub

Logging in to GitHub

You should now be logged in to GitHub! In the future, to log on to GitHub, go to https://github.com/, where you will be presented with the homepage. If you aren't already logged in, click on the "Sign in" link at the top.

Once you've done that, you will see the log in page where you will enter in your username and password that you created earlier.

Sign in to GitHub

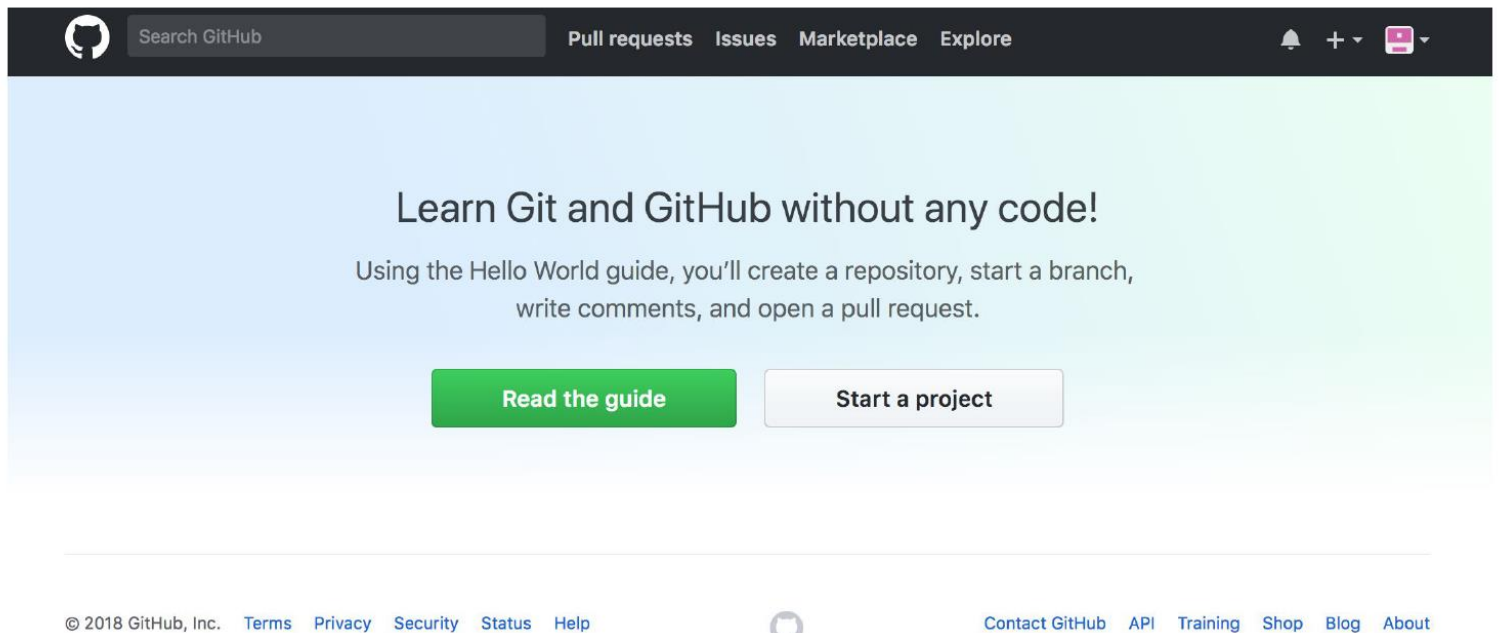Username or email address

Password   Forgot password?

Sign in

New to GitHub? Create an account.

github.com/login

GitHub's log in page

Once logged in, you will be back at https://github.com/, but this time the screen should look like this:
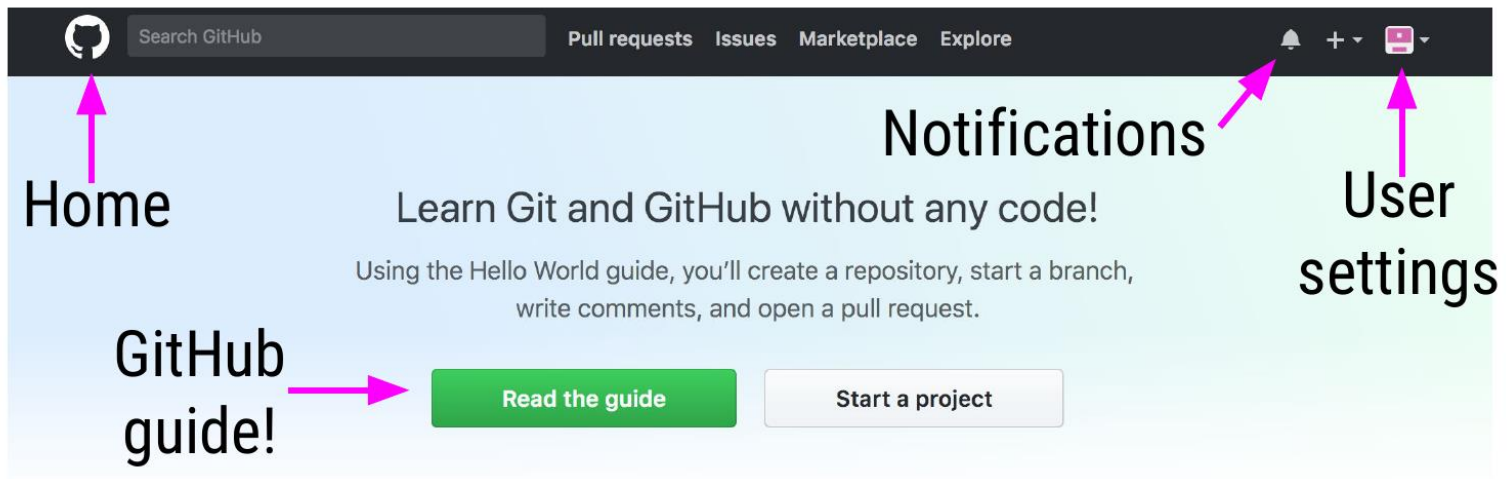
https://qithub.com/

GitHub's homepage at https://github.com/

The homepage

We're going to take a quick tour of the GitHub website, and we'll particularly focus on these sections of the interface:

1. User settings
2. Notifications
3. Help files
4. The GitHub guide

Following this tour, we'll make your very first repository using the GitHub guide!
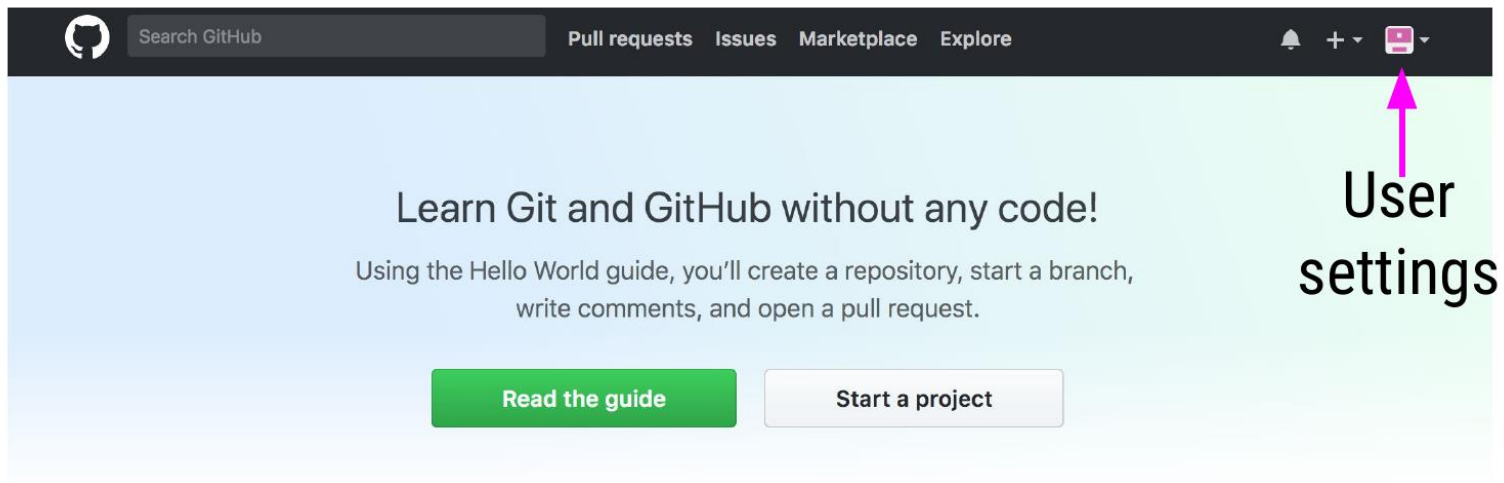
Some major features of GitHub

User settings

Now that you've logged on to GitHub, we should fill out some of your profile information and get acquainted with the account settings. In the upper right corner, there is an icon with an arrow beside it, click this and go to "Your profile"

Where to find user settings

This is where you control your account from and can view your contribution histories and repositories.

Your profile

Since you are just starting out, you aren't going to have any repositories or contributions yet - but hopefully we'll change that soon enough! What we can do right now is edit your profile.

Go to "Edit profile" along the lefthand edge of the page. Here, take some time and fill out your name and a little description of yourself in the "Bio" box, and if you like, upload a picture of yourself! When you are done, click "Update profile"

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

Saved replies

Applications

Developer settings

Public profile

Name

Public email

Select a verified email to display ⇕

You have set your email address to private. To toggle email privacy, go to email settings and uncheck "Keep my email address private."

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

URL

Company

You can @mention your company's GitHub organization to link it.

Location

Update profile

Profile picture

Upload new picture

Editing your profile page

Along the lefthand side of this page, there are many options for you to explore. Click through each of these menus to get familiar with the options available to you. To get you started, go to the account page.

Your account page

Here, you can edit your password or if you are unhappy with your username, change it. Be careful though, there can be unintended consequences when you change your username - if you are just starting out and don't have any content yet, you'll probably be safe though.

Continue looking through the personal setting options on your own. When you are done, go back to your profile.

Once you've had a bit more experience with GitHub, you'll eventually end up with some repositories to your name. To find those, click on the "Repositories" link on your profile. For now, it will probably look like this:
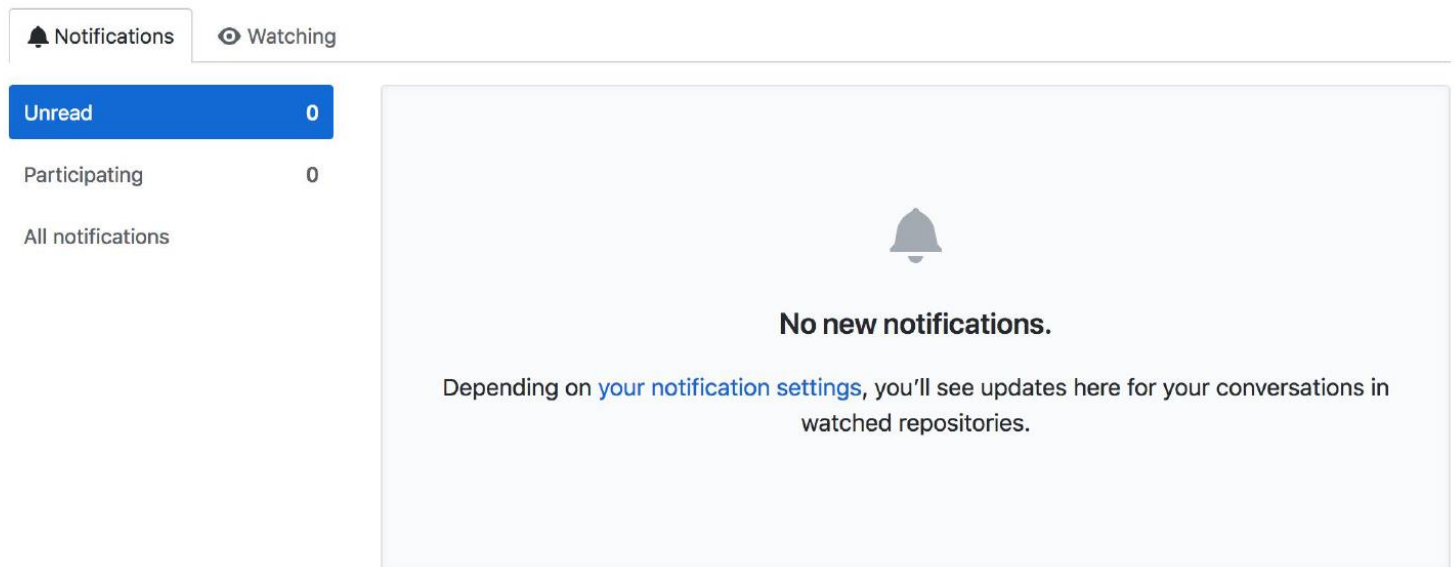
Your repositories page

By the end of the lecture though, check back to this page to find your newly created repository!

Notifications

Next, we'll check out the notifications menu. Along the menu bar across the top of your window, there is a bell icon, representing your notifications. Click on the bell.
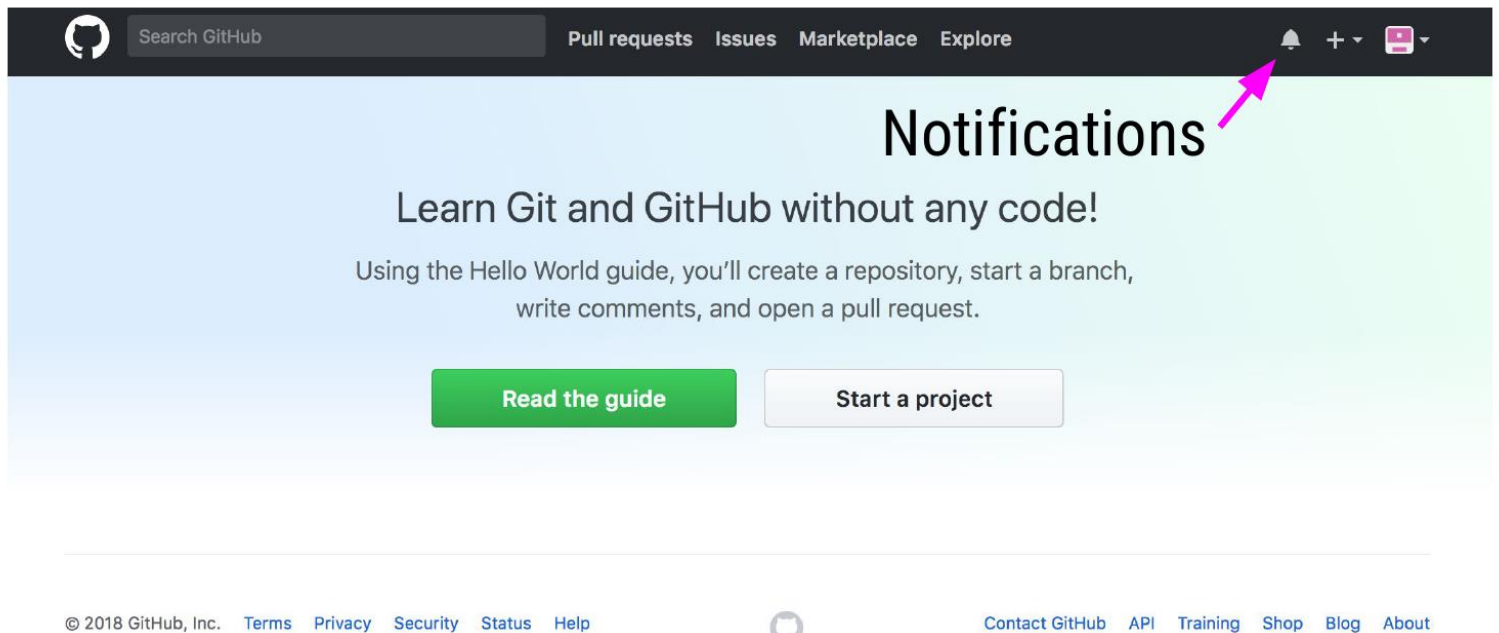
Location of the bell icon

Your notifications

Once you become more active on GitHub and are collaborating with others, here is where you can find messages and notifications for all the repositories, teams, and conversations you are a part of.

Help files

Along the bottom of *every. single. page.* there is the [“Help” button](#). GitHub has a great help system in place - if you ever have a question about GitHub, this should be your first point to search! Take some time now and look through the various help files, and see if any catch your eye.
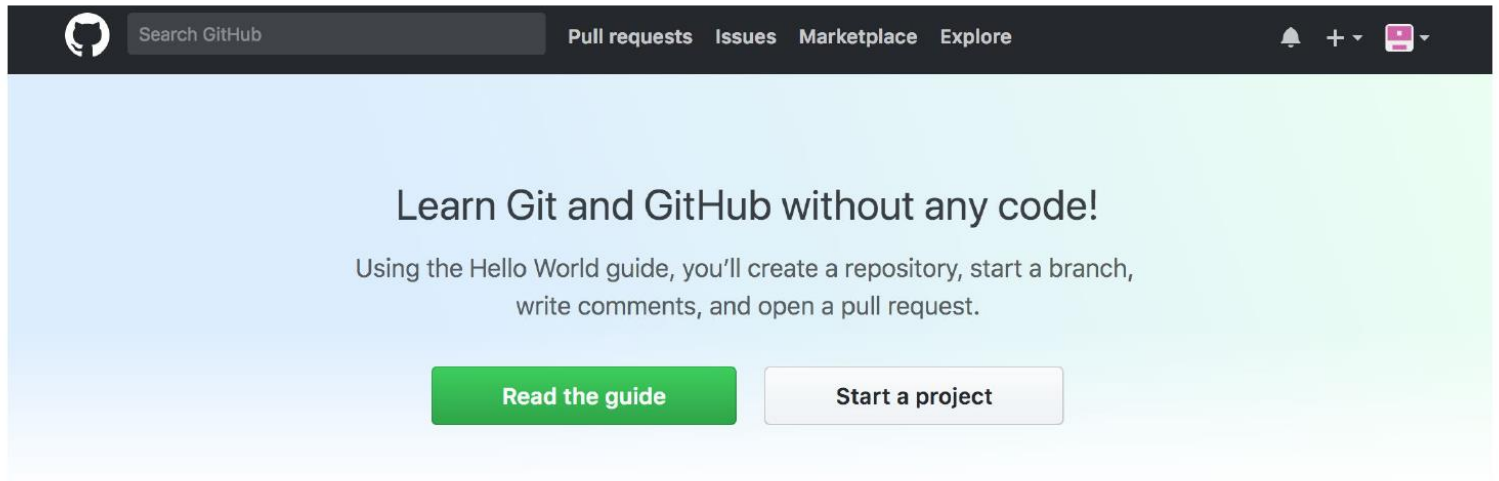
At the bottom of every page, you can find the Help page

Sometimes you just need a little help.

| How can we help? | 🔍 |

Bootcamp

> Set Up Git
> Create A Repo
> Fork A Repo
> Be Social

Setup

> Signing up for a new GitHub account
> Verifying your email address
> About commit email addresses
> Setting your commit email address on GitHub
> Setting your commit email address in Git
> Blocking command line pushes that expose your personal email address
> Setting your username in Git
> Dealing with line endings
> Supported browsers

https://help.github.com/
GitHub's help files

The GitHub guide

GitHub recognizes that this can be an overwhelming process for new users, and as such have developed a mini tutorial to get you started with GitHub. Go through this guide now and create your first repository! When you are done, you should have a repository that looks something like this:

JaneEverydayDoe / **hello-world**

👁 Watch ▾ 0   ★ Star 0   ⑂ Fork 0

<> Code   ⊙ Issues 0   ⎇ Pull requests 0   ▥ Projects 0   ▣ Wiki   ▥ Insights   ⚙ Settings

My first repository!   [Edit]

Add topics

⊙ **3 commits**   ⑁ **1 branch**   ◇ **0 releases**   ⚏ **1 contributor**

Branch: master ▾   New pull request   Create new file Upload files Find file   **Clone or download** ▾

⬛ **JaneEverydayDoe** Merge pull request #1 from JaneEverydayDoe/readme-edits ⋯   Latest commit c9a6e3e a minute ago

📄 README.md   Updating README with location of instructions   3 minutes ago

▤ **README.md**

hello-world

My first repository!

Here is my first repository on this account. I created it entirely within the GitHub interface using the instructions located at:
https://guides.github.com/activities/hello-world

─────────────────○─────────────────

Your first repository

Take some time to explore around the repository - Check out your commit history so far. Here you can find all of the changes that have been made to the repository, and you can see **who** made the change, **when** they made the change, and provided you wrote an appropriate commit message, you can see **why** they made the change! It should look like similar to this:

Your first repository's commit history

Once you've explored all of the options in the repository, go back to your user profile. It should look a little different from before:

Your profile now shows your first repository

Now when you are on your profile you can see your latest repository created and for a complete listing of your repositories, click on the "Repositories" tab. Here you can see all of your repositories, a brief description, the time of the last edit, and along the right hand side, there is an activity graph, showing when and how many edits have been made on the repository.

Your shiny new repository page!

Git

As you may remember from our last lecture, Git is the free and open source version control system which GitHub is built on.

One of the main benefits of using the Git system is its compatibility with RStudio; however, in order to link the two software together, we first need to download and install Git on your computer.

Downloading and installing Git

To download Git, go to https://git-scm.com/download. You should arrive at a webpage like this:
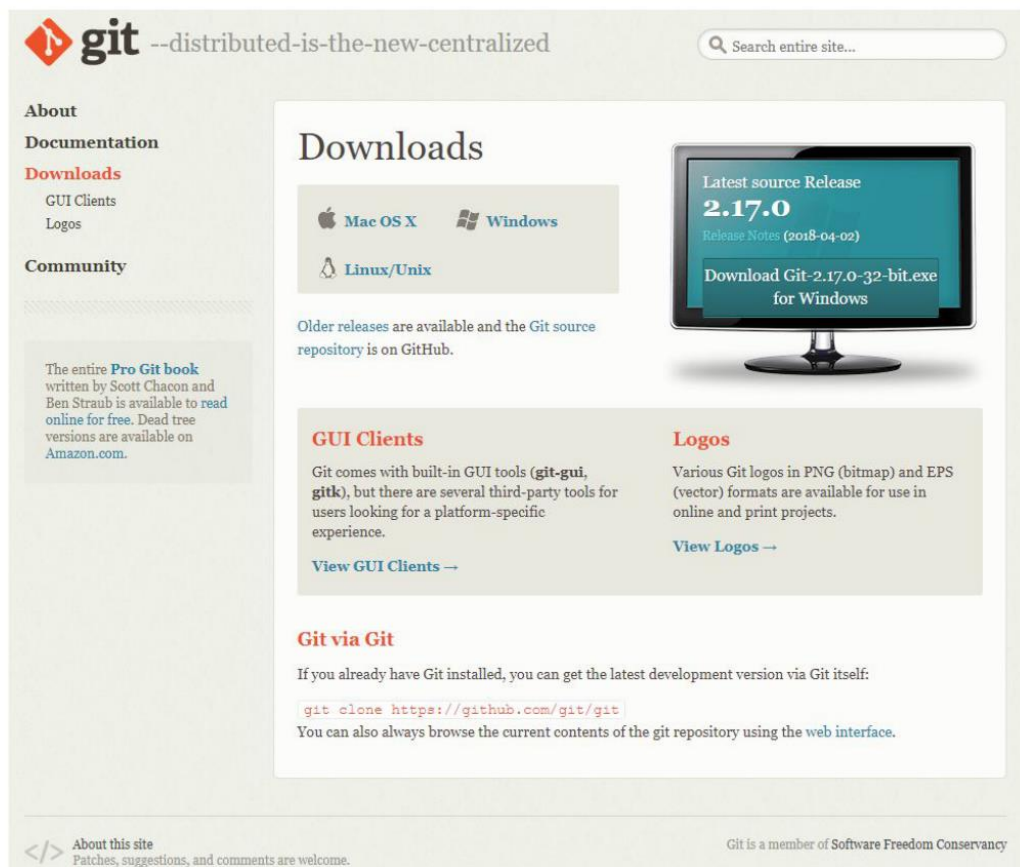
Downloading Git from git-scm.com/download

Click on the appropriate download link for your operating system. This should initiate the download process.

For Windows

Once the download is finished, open the .exe file to initiate the installation wizard. If you receive a security warning, click "Run" and/or "Allow." Following this, click through the installation wizard, generally accepting the default options unless you have a compelling reason not to.
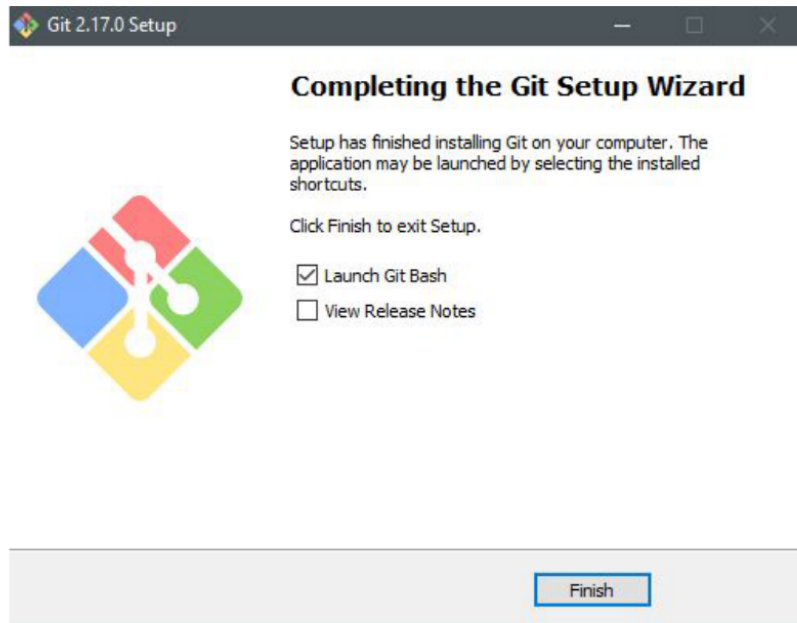
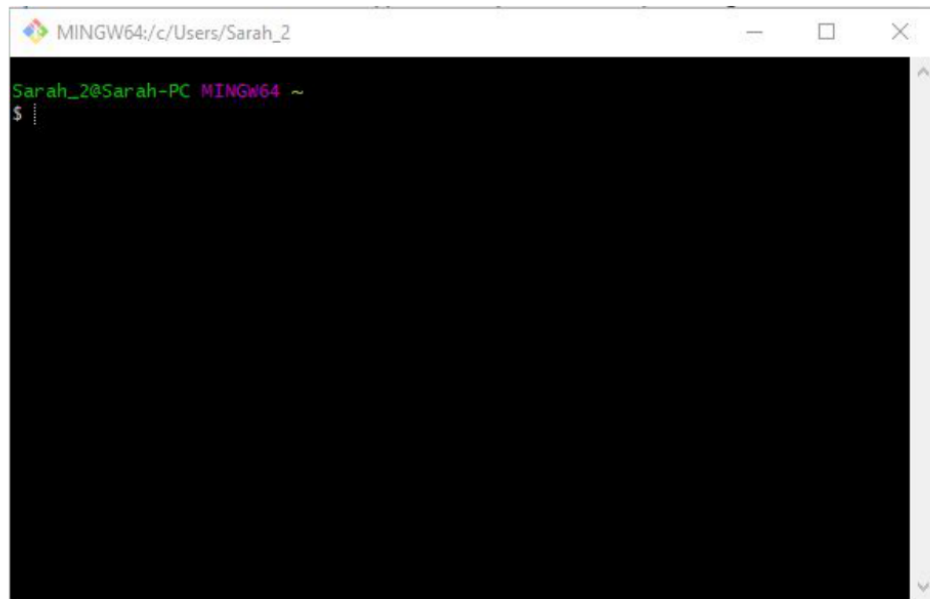# Windows installation process
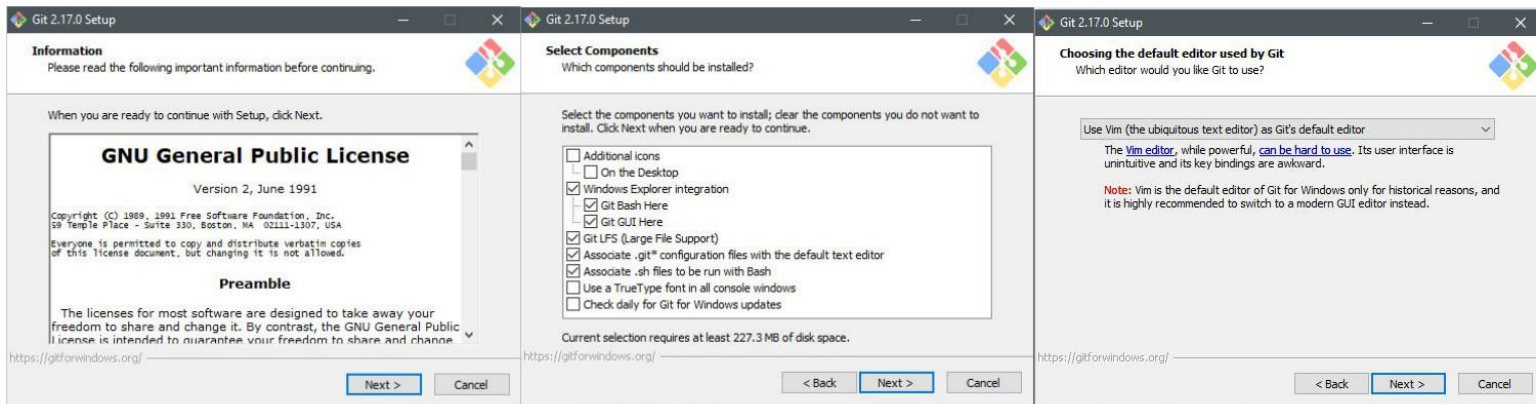
… and so on

Installation wizard for Git on Windows

Click "Install" and allow the wizard to complete the installation process. Following this, check the "Launch Git Bash" option, and unless you are curious, deselect the "View Release Notes" box, as you are probably not interested in this right now.

Finishing the install process

Doing so, a command line environment will open. Provided you accepted the default options during the installation process, there will now be a Start menu shortcut to launch Git Bash in the future. You have now installed Git.

MINGW64:/c/Users/Sarah_2

Sarah_2@Sarah-PC MINGW64 ~
$

---○---

Git Bash is the command line interface you will use to configure Git

For Mac

We will walk you through the most common installation process however, there are multiple ways to get Git onto your Mac. You can follow the tutorials at https://www.atlassian.com/git/tutorials/install-git for alternative installation routes.

After downloading the appropriate Git version for Macs, you should have downloaded a DMG file for installation on your Mac. Open this file. This will install Git on your computer. A new window will open.

Macs installation process
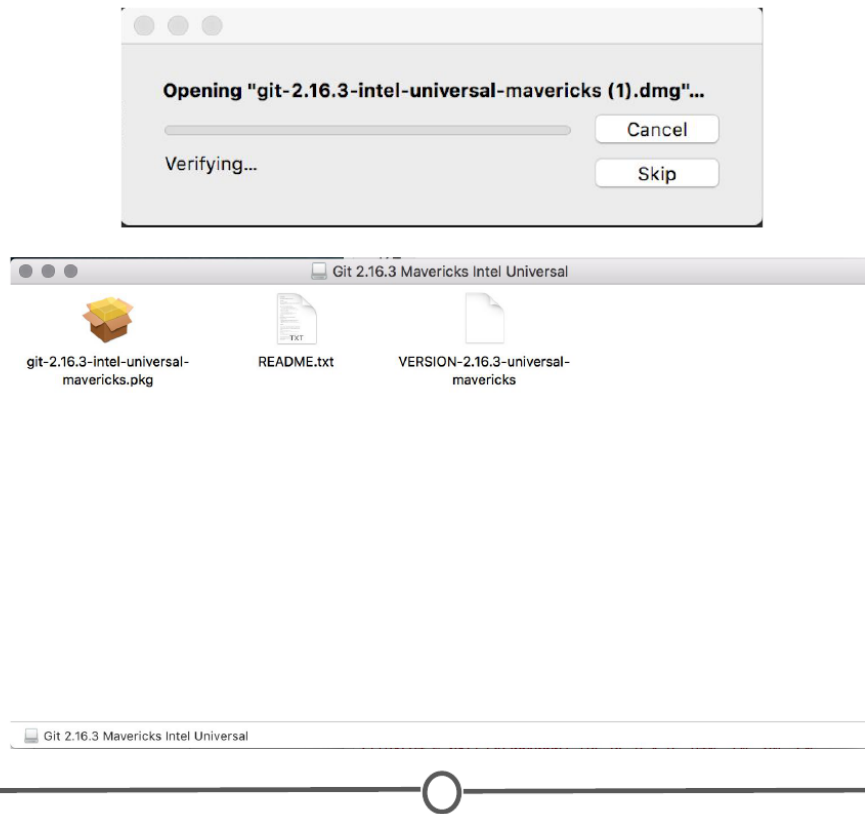
Installation wizard for Git on Mac

Double click on the .pkg file and an installation wizard will open. Click through the options, accepting the defaults. Click Install. When prompted, close the installation wizard. You have successfully installed Git!
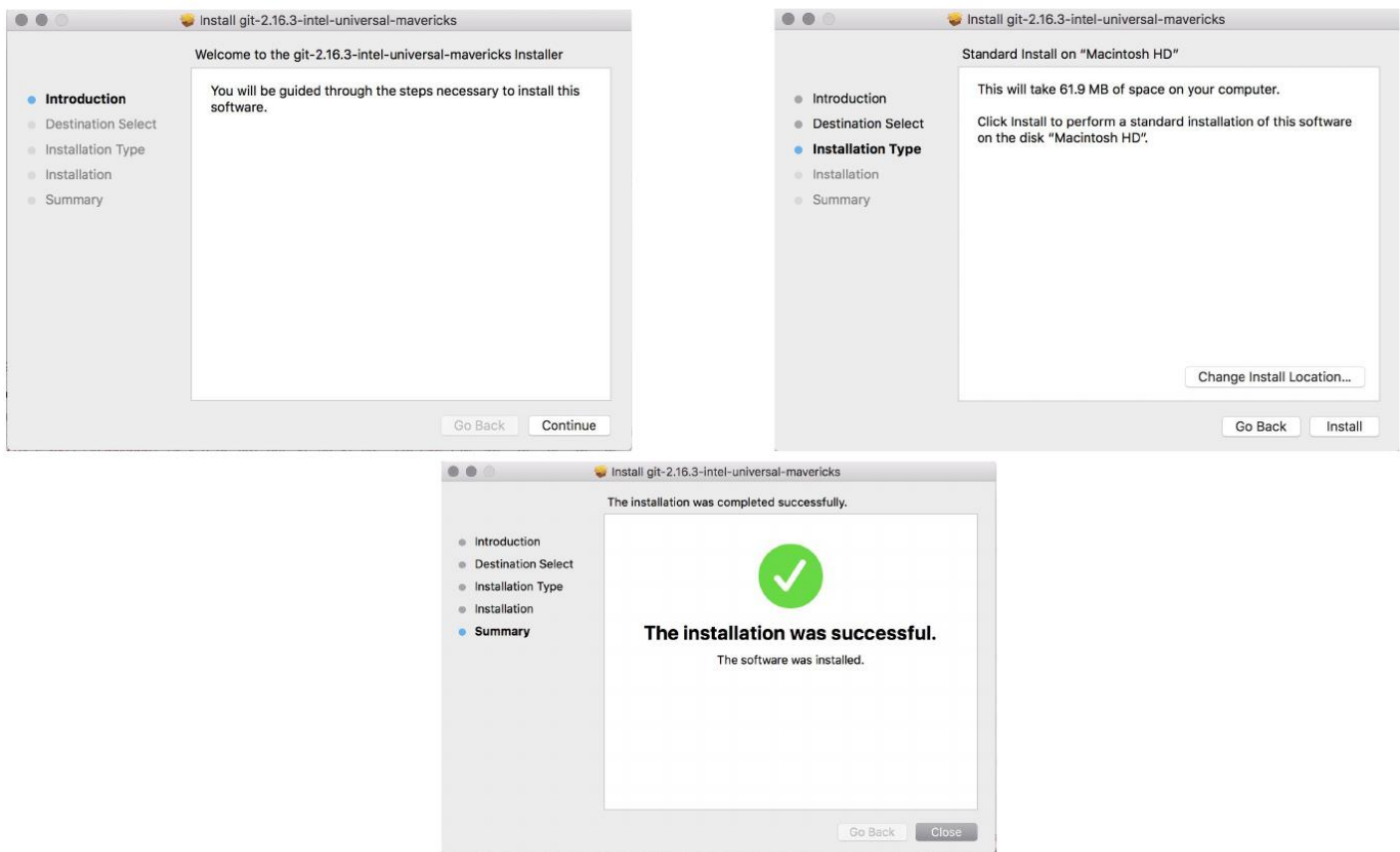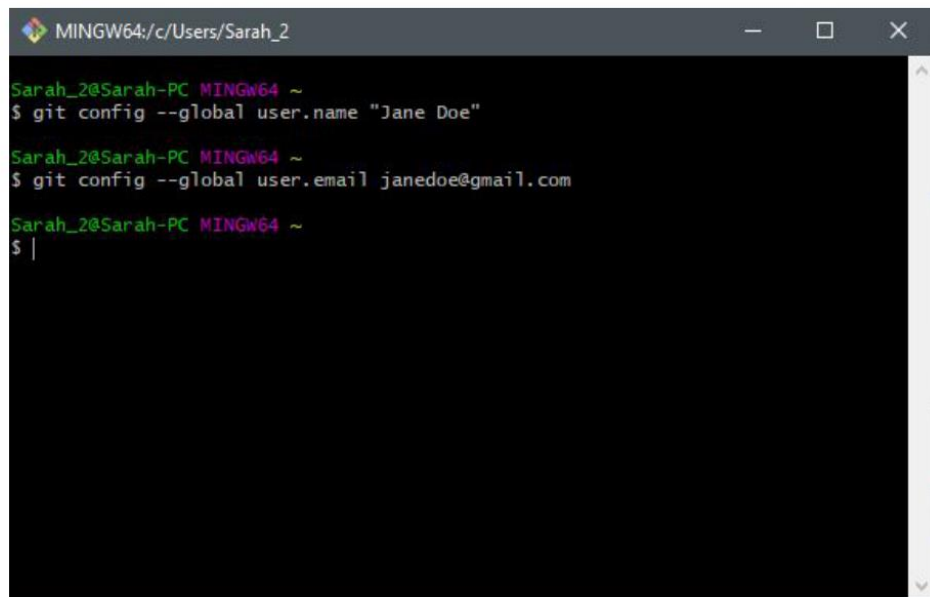
Steps to a successful installation of Git!

Configuring Git

Now that Git is installed, we need to configure it for use with GitHub, in preparation for linking it with RStudio.

We need to tell Git what your username and email are, so that it knows how to name each commit as coming from you. To do so, in the command prompt (either Git Bash for Windows or Terminal for Mac), type: `git config --global user.name "Jane Doe"` with your desired username in place of "Jane Doe." This is the name each commit will be tagged with.

Following this, in the command prompt, type: `git config --global user.email janedoe@gmail.com` **MAKING SURE TO USE THE SAME EMAIL ADDRESS YOU SIGNED UP FOR GITHUB WITH!**

Configuring Git to tag each commit with your name and interface with GitHub

Confirming your configuration

At this point, you should be set for the next step, but just to check, confirm your changes by typing: `git config --list`

Confirming your user name and user email

Doing so, you should see the username and email you selected above. If you notice any problems or want to change these values, just retype the original `config` commands from earlier with your desired changes.

Once you are satisfied that your username and email is correct, exit the command line by typing `exit` and hit Enter. At this point, you are all set up for the next lecture!

Exiting the command prompt

Summary

In this lesson, we signed up for a GitHub account and toured the GitHub website. We made your first repository and filled in some basic profile information on GitHub. Following this, we installed Git on your computer and configured it for compatibility with GitHub and RStudio.