GitHub and Git

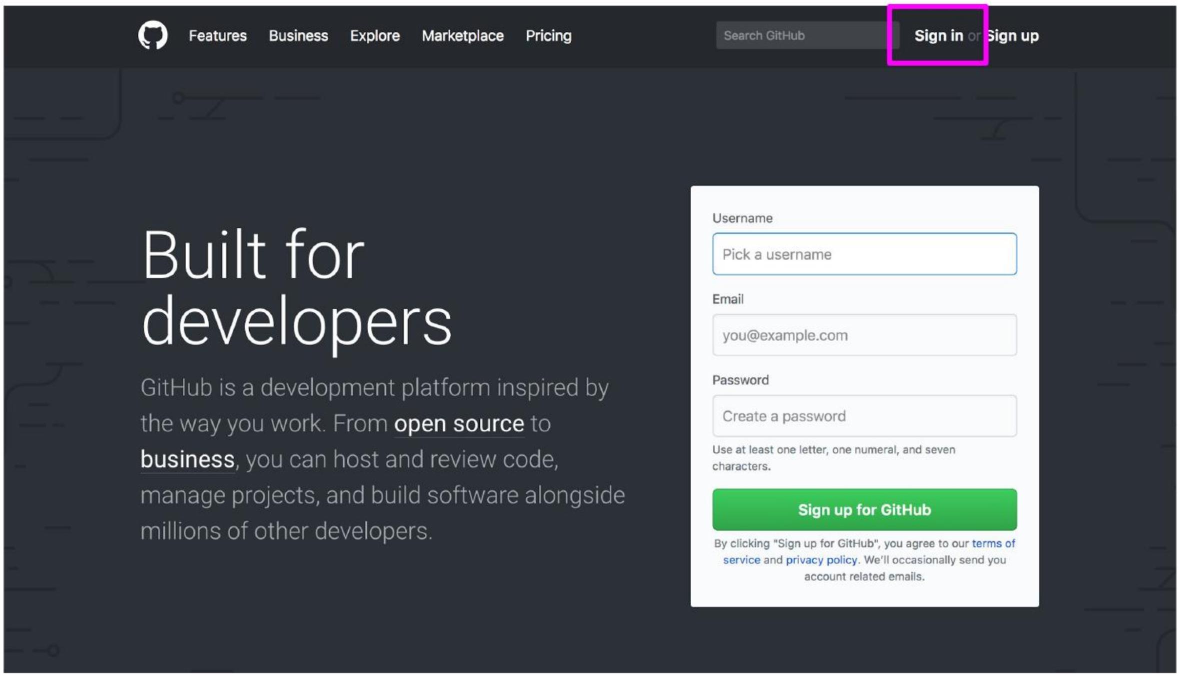
Now that we've got a handle on what version control is, in this lesson, you will sign-up for a GitHub account, navigate around the GitHub website to become familiar with some of its features, and install and configure Git; all in preparation for linking both with your RStudio!

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."



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside millions of other developers.

Use at least one letter, one numeral, and seven characters.

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.

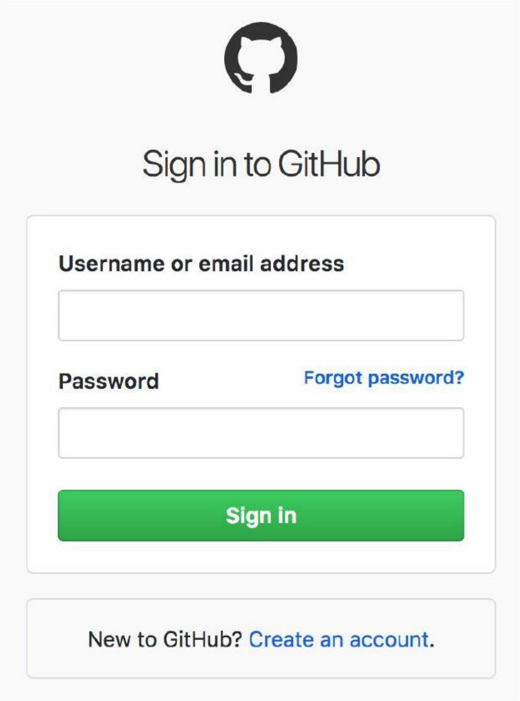
github.com

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.

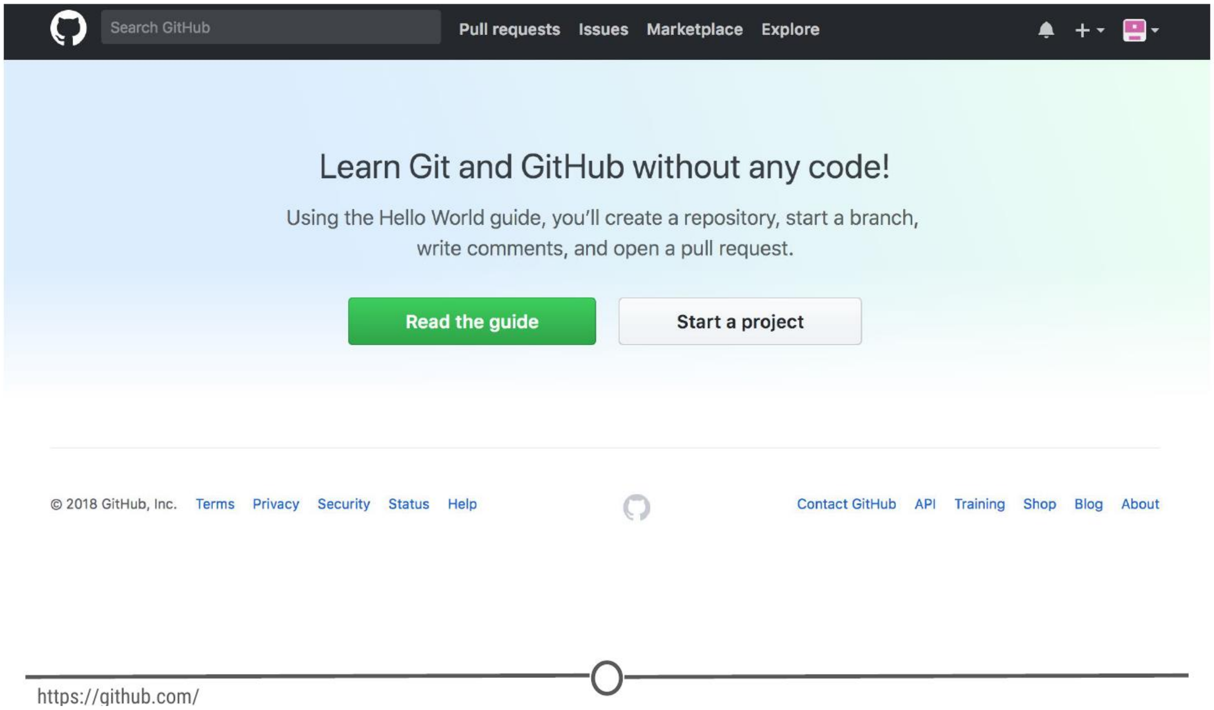


New to GitHub? Create an account.

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:



Learn Git and GitHub without any code!

Using the Hello World guide, you’ll create a repository, start a branch, write comments, and open a pull request.

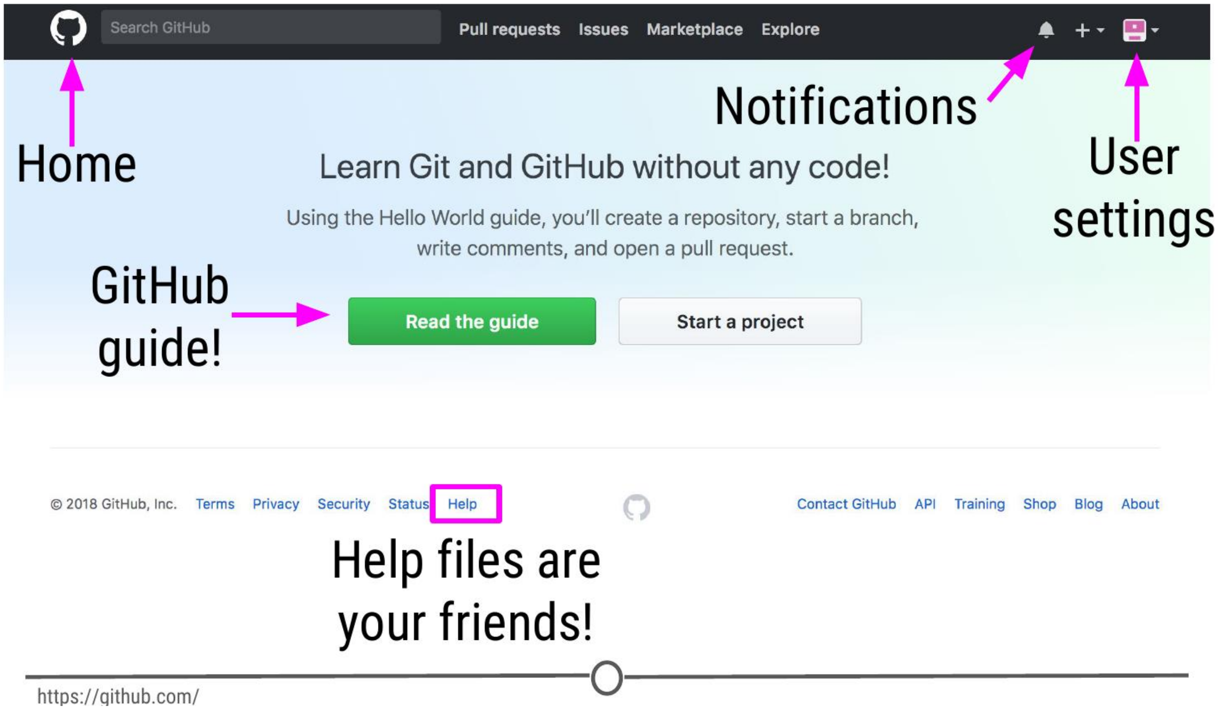
GitHub’s homepage at https://github.com/

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!



Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Home

Notifications

User settings

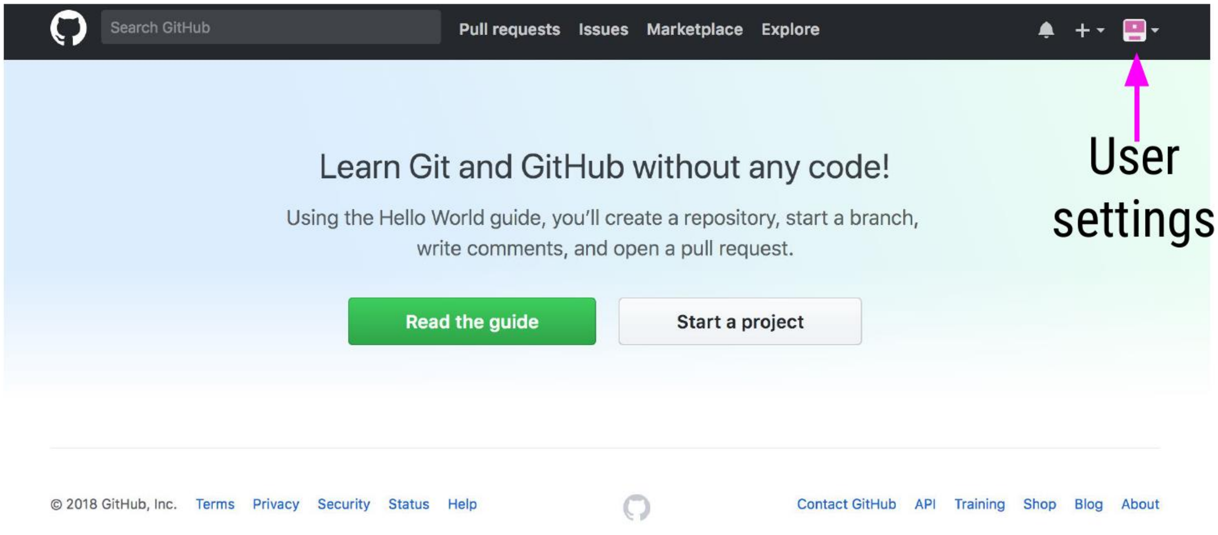
GitHub guide!

Help files are your friends!

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"



Learn Git and GitHub without any code!

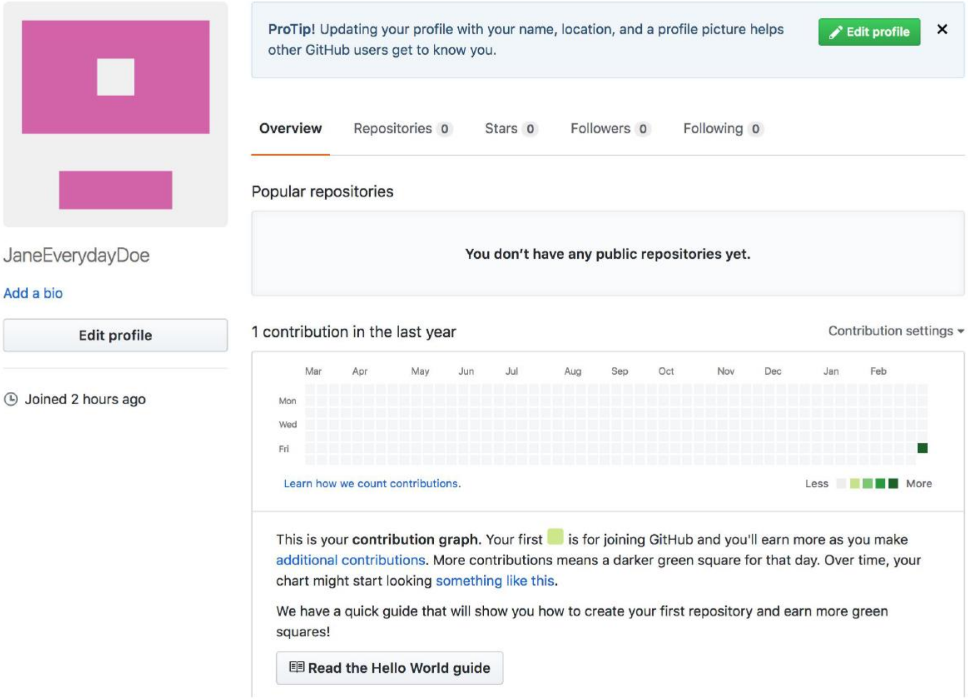
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

© 2018 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub API Training Shop Blog About

https://github.com/

Where to find user settings

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



ProTip! Updating your profile with your name, location, and a profile picture helps other GitHub users get to know you.

You don't have any public repositories yet.

1 contribution in the last year

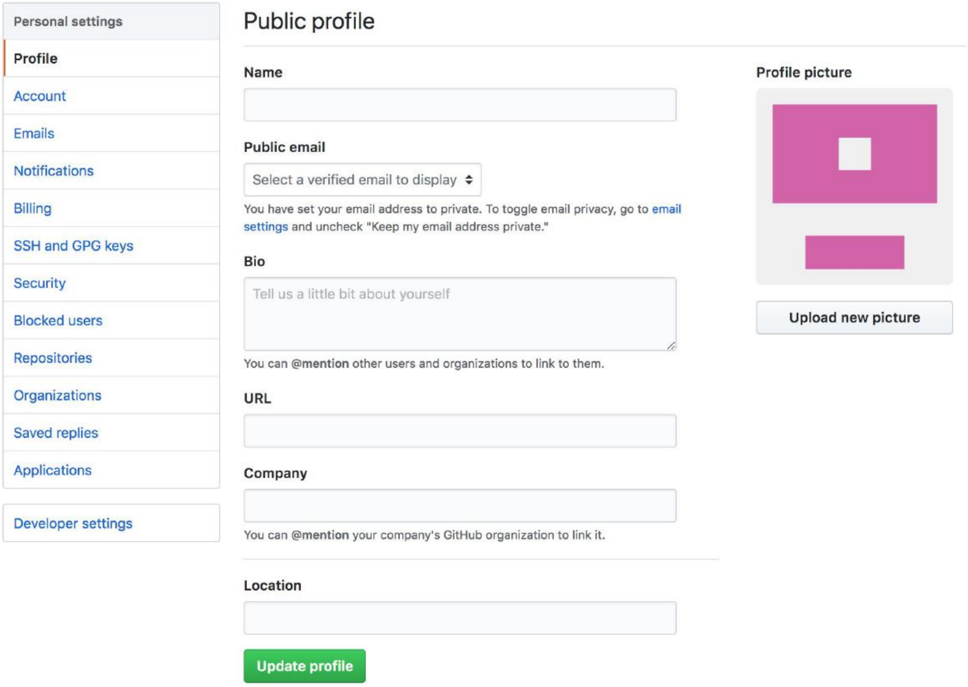
This is your **contribution graph**. Your first 🟩 is for joining GitHub and you'll earn more as you make additional contributions. More contributions means a darker green square for that day. Over time, your chart might start looking something like this.

We have a quick guide that will show you how to create your first repository and earn more green squares!

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

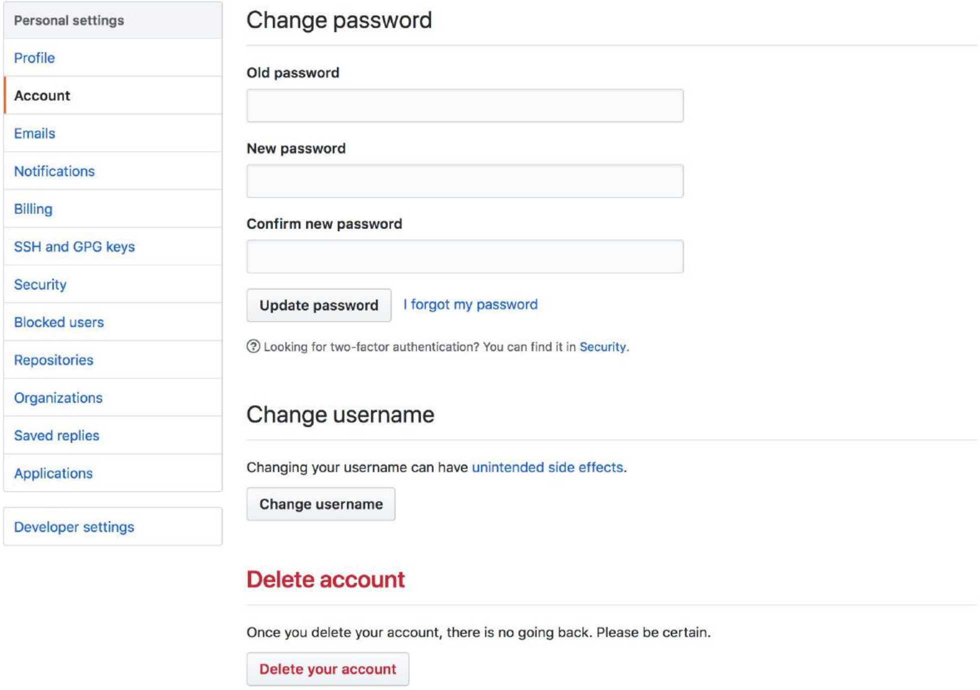
Update profile

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.



Change password

Old password

New password

Confirm new password

Looking for two-factor authentication? You can find it in Security.

Change username

Changing your username can have unintended side effects.

Delete account

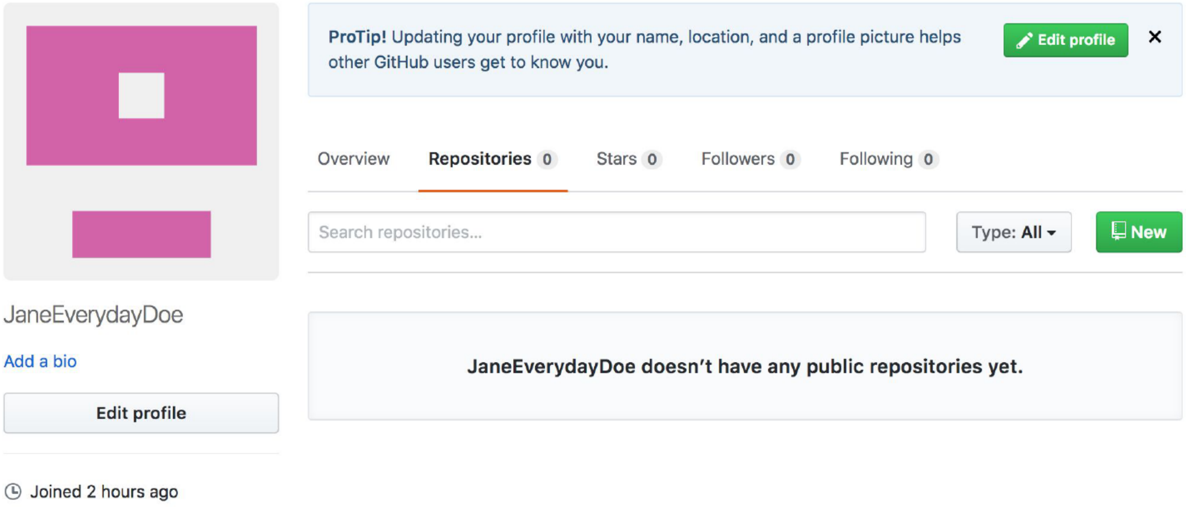
Once you delete your account, there is no going back. Please be certain.

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:



ProTip! Updating your profile with your name, location, and a profile picture helps other GitHub users get to know you.

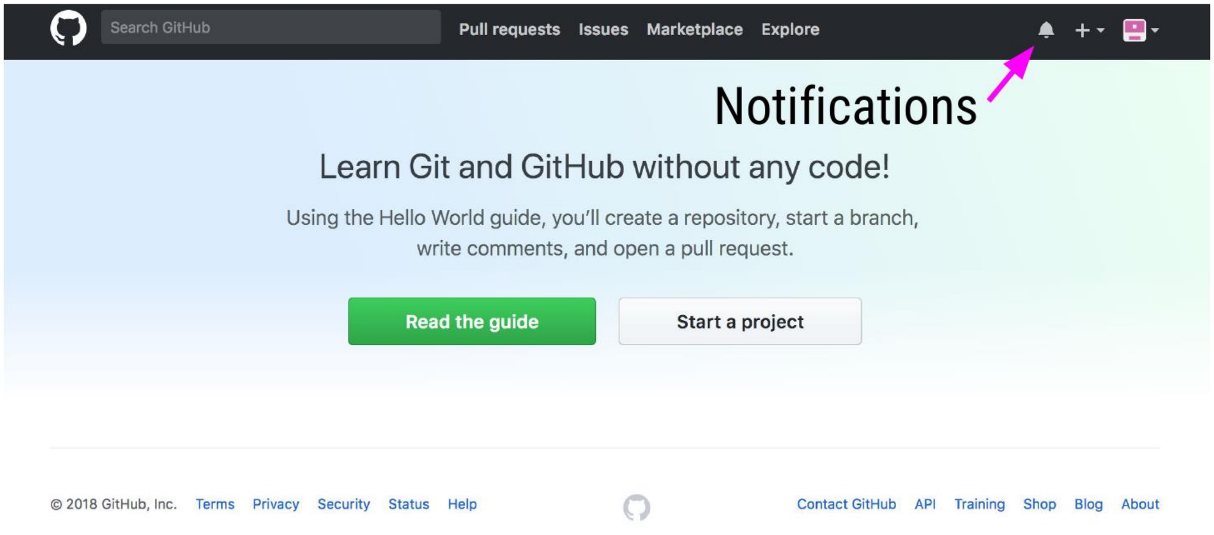
JaneEverydayDoe doesn't have any public repositories yet.

Your repositories page

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

Notifications

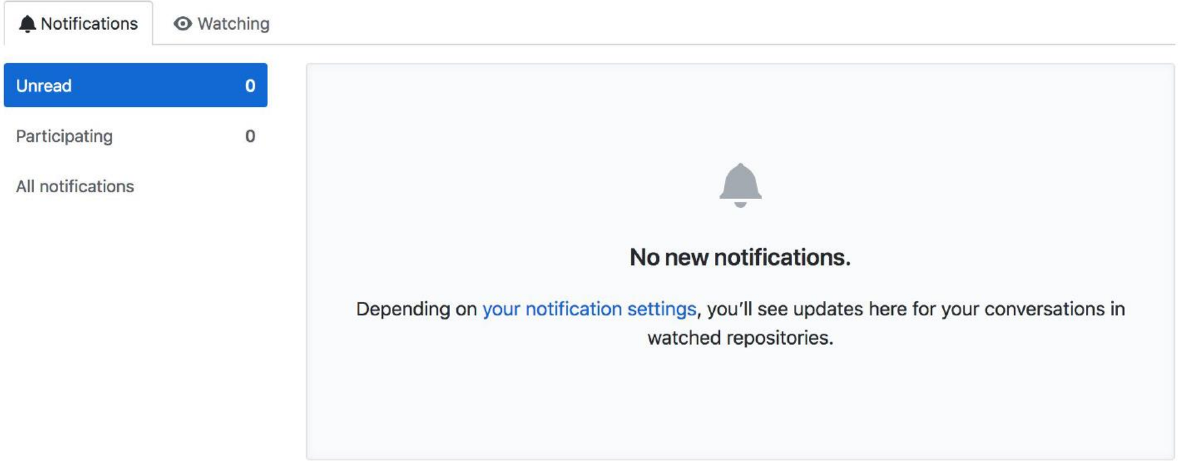
Next, we'll check out the notifications menu. Along the menu bar across the top of your window, there is a bell icon, representing your notifications. Click on the bell.



Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Location of the bell icon



No new notifications.

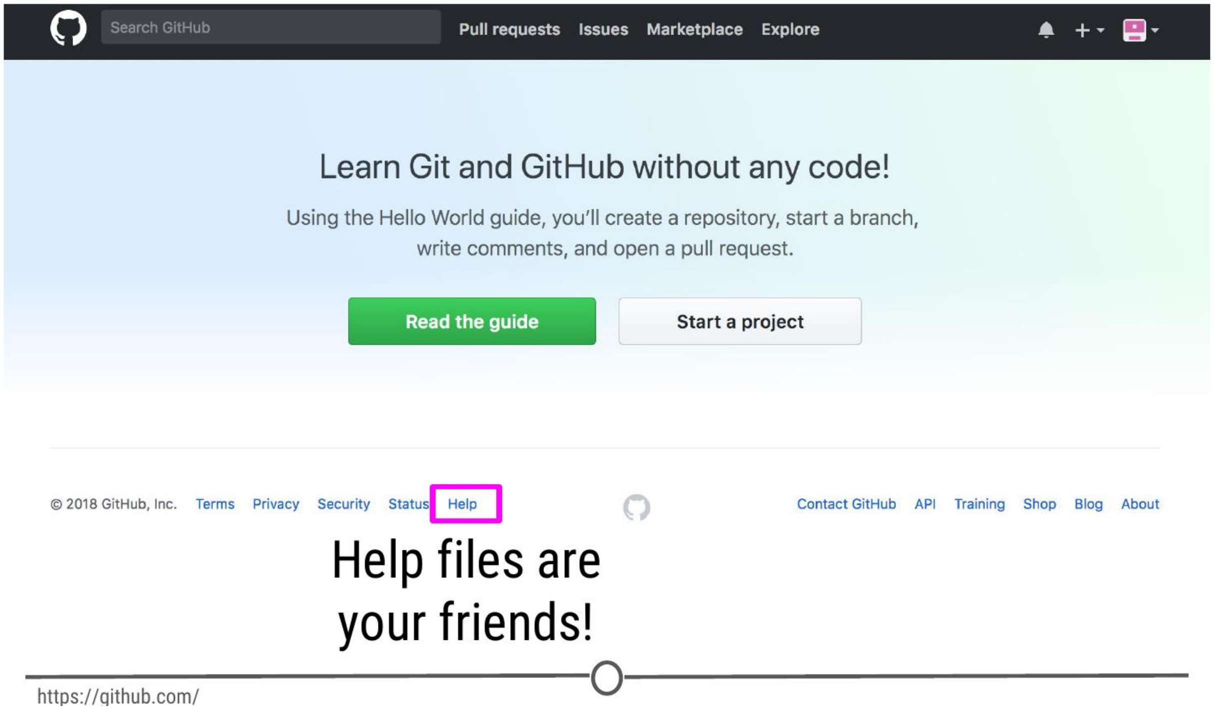
Depending on your notification settings, you'll see updates here for your conversations in watched repositories.

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

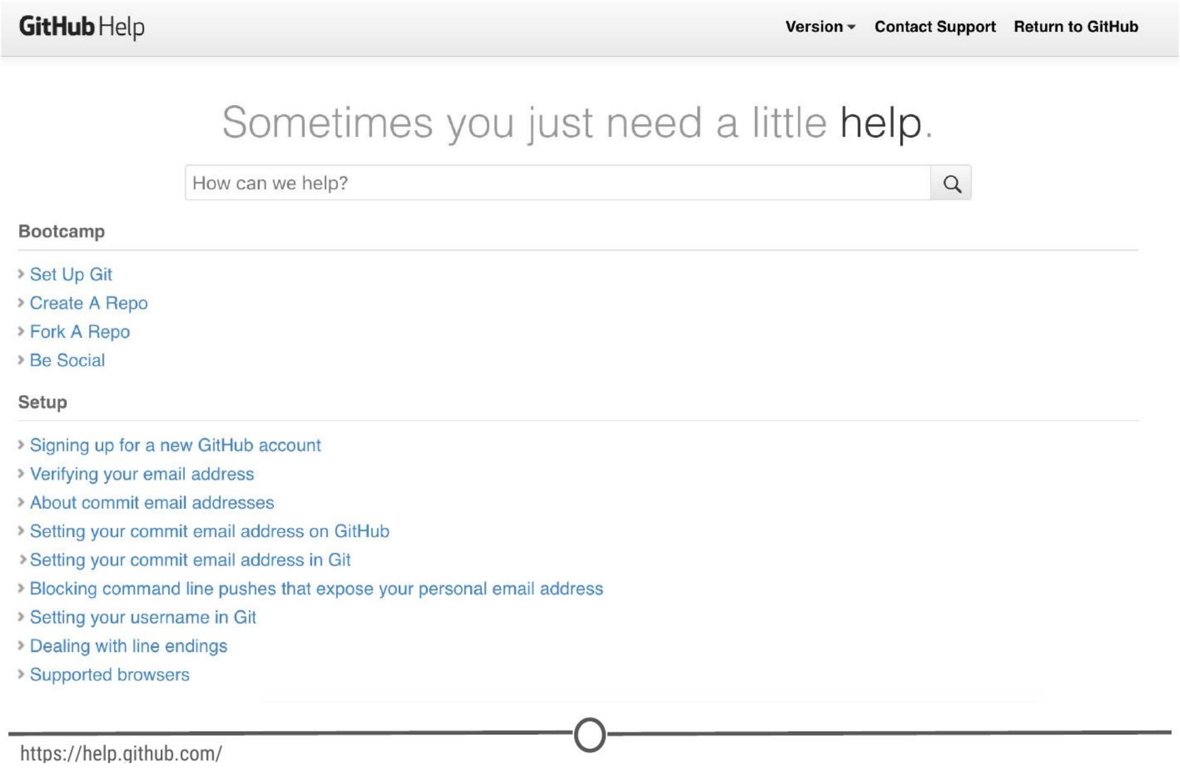
Along the bottom of *every. single. page.* there is the [“Help” button](). GitHub has a great help system in place - if you ever have a question about GitHub, this should be your first point to search! Take some time now and look through the various help files, and see if any catch your eye.



Learn Git and GitHub without any code!

Using the Hello World guide, you’ll create a repository, start a branch, write comments, and open a pull request.

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

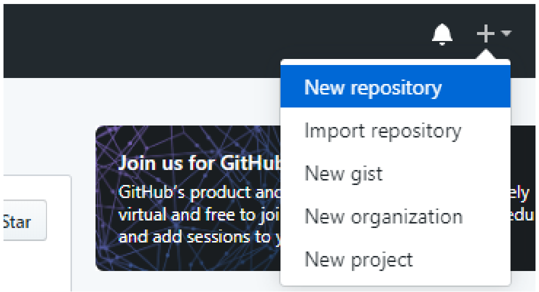
- Signing up for a new GitHub account
- Verifying your email address
- About commit email addresses
- Setting your commit email address on GitHub
- Setting your commit email address in Git
- Blocking command line pushes that expose your personal email address
- Setting your username in Git
- Dealing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

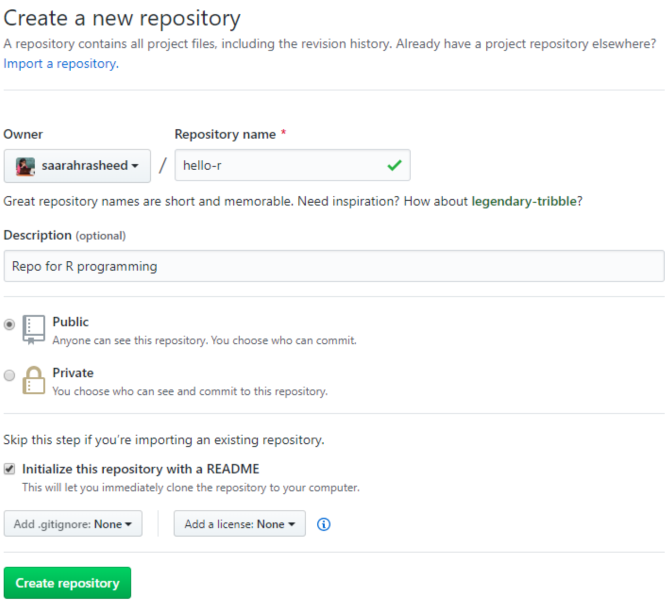
https://help.github.com/

GitHub's help files

Creating a Repository

To create a repository, go to Homepage > Your profile > Repositories > Create New. Alternatively, you can click on the little + sign at the top right corner to create a repository easily.





Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: saarahrasheed ▾ / Repository name *: hello-r ✓

Great repository names are short and memorable. Need inspiration? How about legendary-tribble?

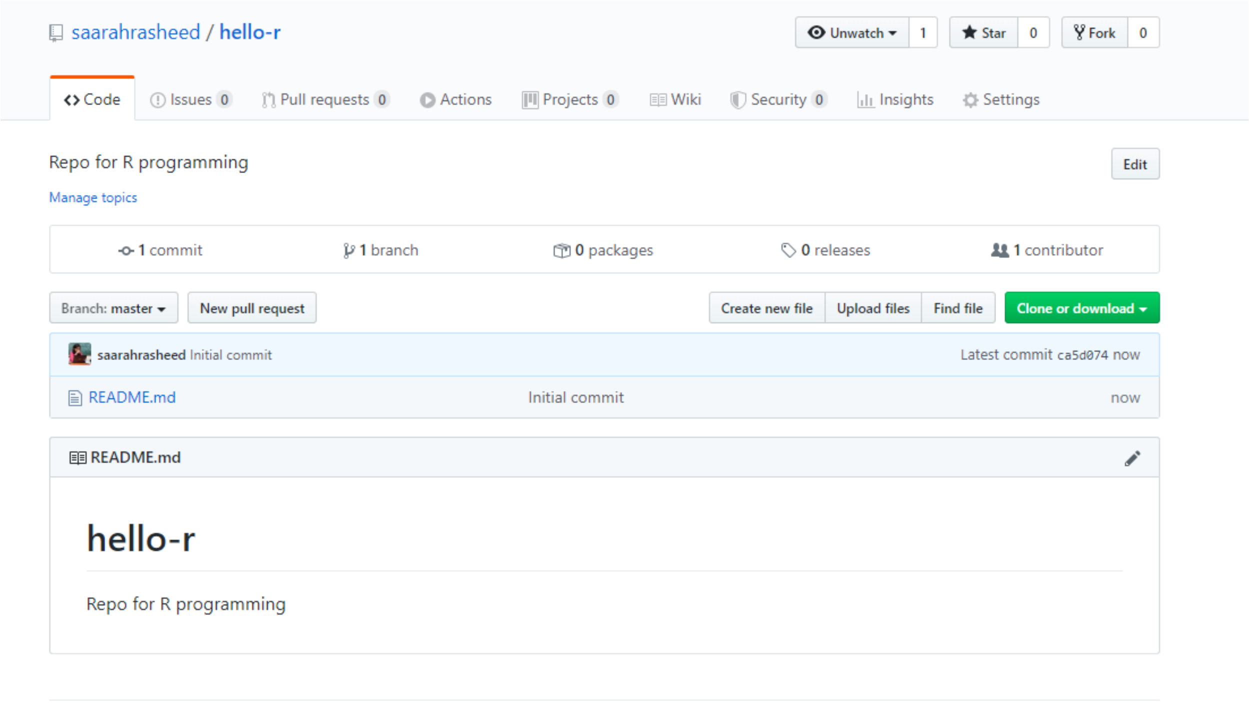
Description (optional): Repo for R programming

● Public
Anyone can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☑ Initialize this repository with a README
This will let you immediately clone the repository to your computer.



Repo for R programming

saarahrasheed Initial commit — Latest commit ca5d074 now

README.md — Initial commit — now

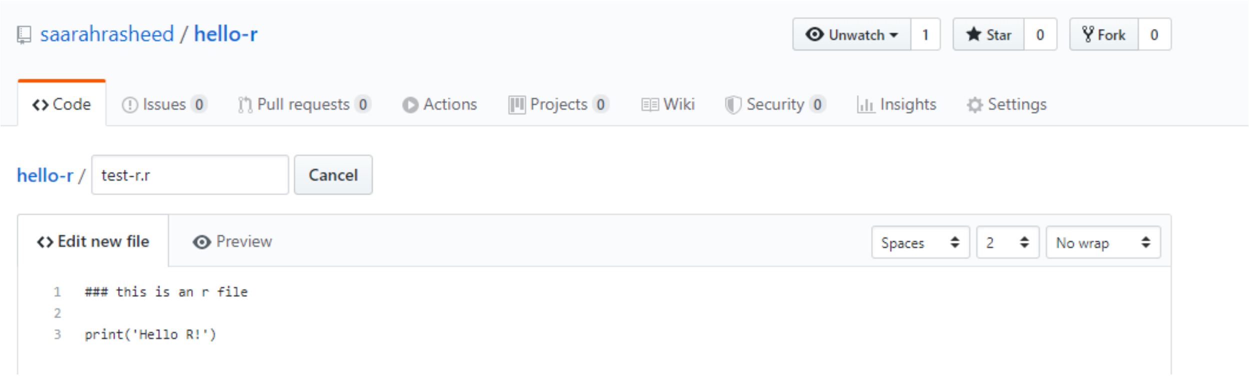
README.md

hello-r

Repo for R programming

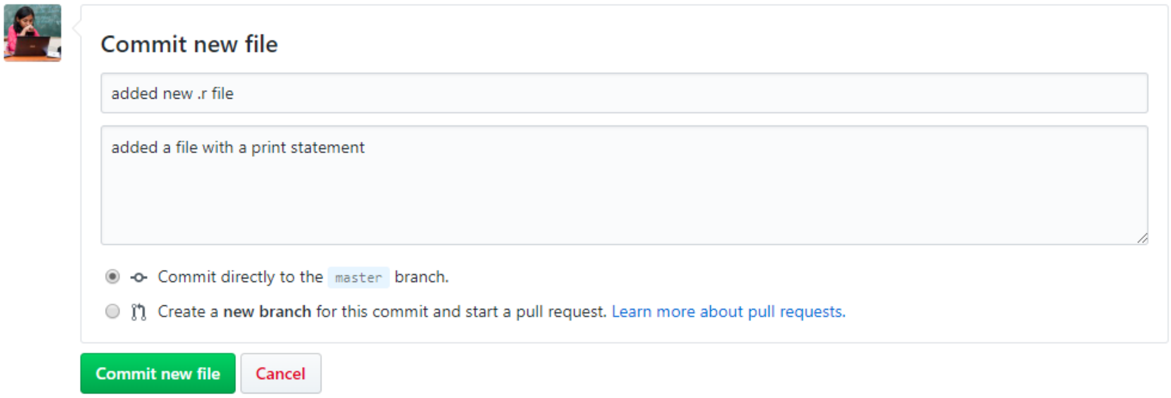
Your repo is ready. Good job!

Let’s try to add some files to it. Click on the create new file. Add a name (since we are working with r, we go with .r extension). Add some code to the file. Once you are done adding code, scroll to the bottom and you find commit section. When you are satisfied with your code, you can add some message and then commit the file to the repo.



hello-r / test-r.r

```
### this is an r file

print('Hello R!')
```



Commit new file

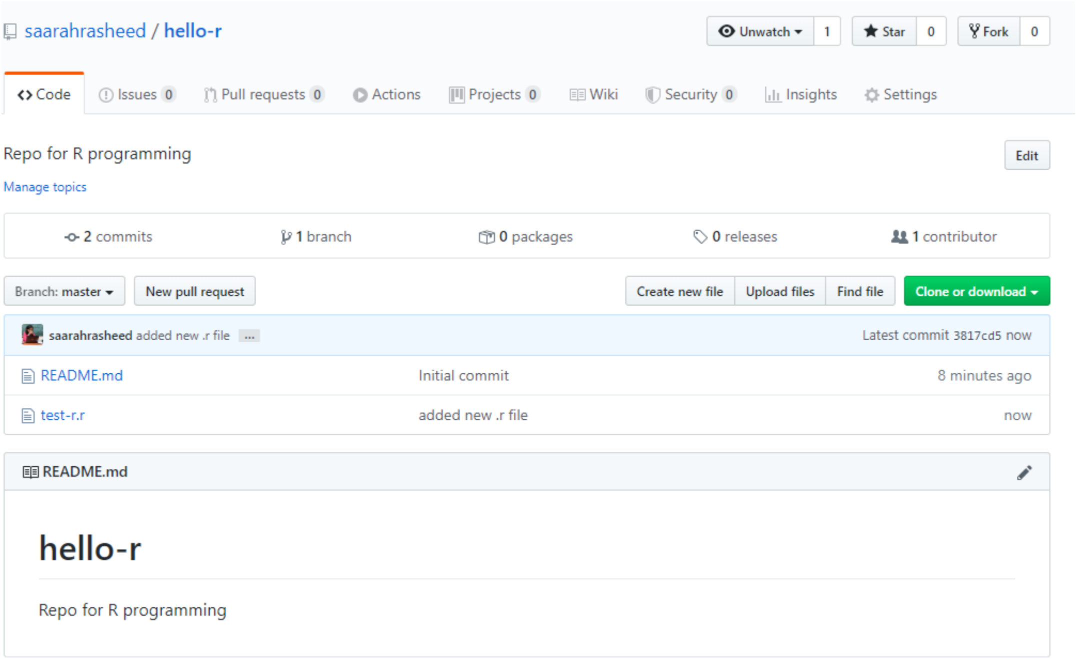
added new .r file

added a file with a print statement

● Commit directly to the `master` branch.

○ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

Congrats! You created your first file in the repo!



Repo for R programming

2 commits · 1 branch · 0 packages · 0 releases · 1 contributor

saarahrasheed added new .r file … Latest commit 3817cd5 now

README.md	Initial commit	8 minutes ago
test-r.r	added new .r file	now

README.md

hello-r

Repo for R programming

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.

Check your shiny new repository page!

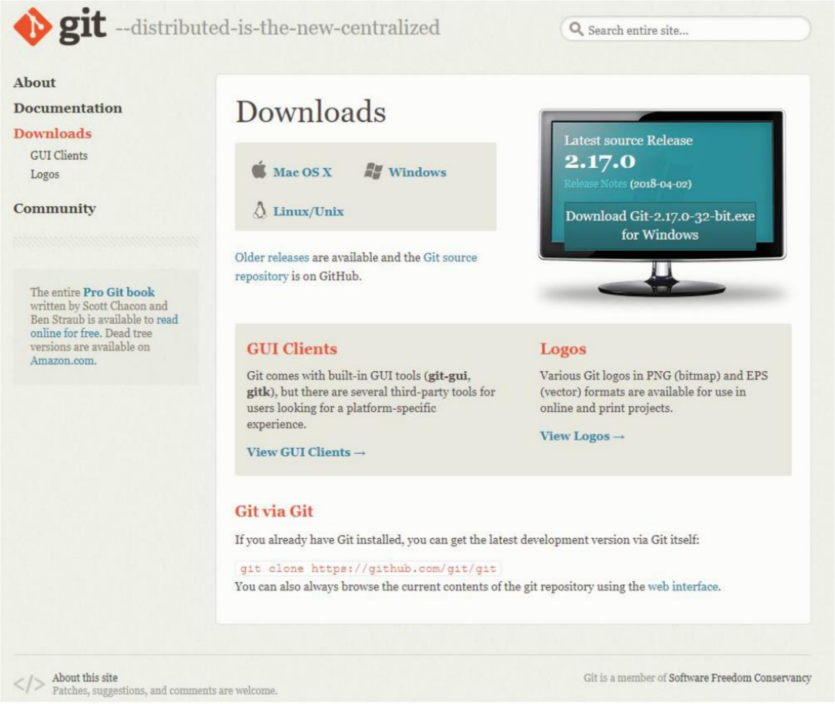
Git

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:



The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Older releases are available and the Git source repository is on GitHub.

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the web interface.

Patches, suggestions, and comments are welcome.

Git is a member of Software Freedom Conservancy

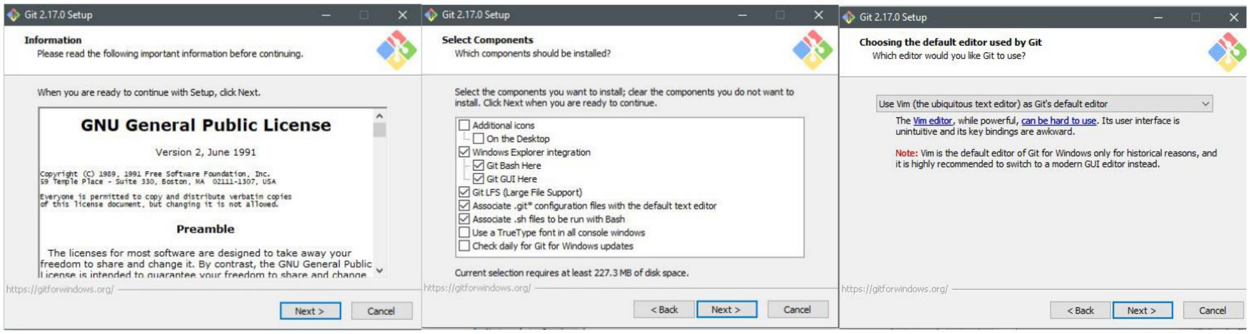
Downloading Git from git-scm.com/download

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



Information
Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

https://gitforwindows.org/

Select Components
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- □ Additional icons
  - □ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Git Bash Here
  - ☑ Git GUI Here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- □ Use a TrueType font in all console windows
- □ Check daily for Git for Windows updates

Current selection requires at least 227.3 MB of disk space.

https://gitforwindows.org/

Choosing the default editor used by Git
Which editor would you like Git to use?

Use Vim (the ubiquitous text editor) as Git's default editor

The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.

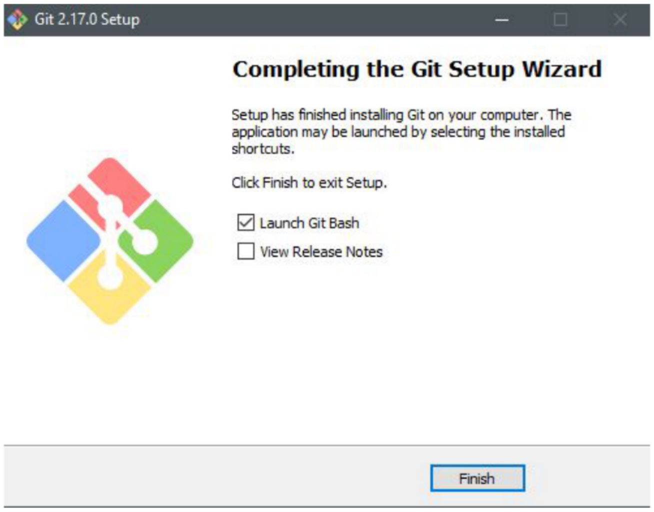
Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.

https://gitforwindows.org/

… 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.



Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

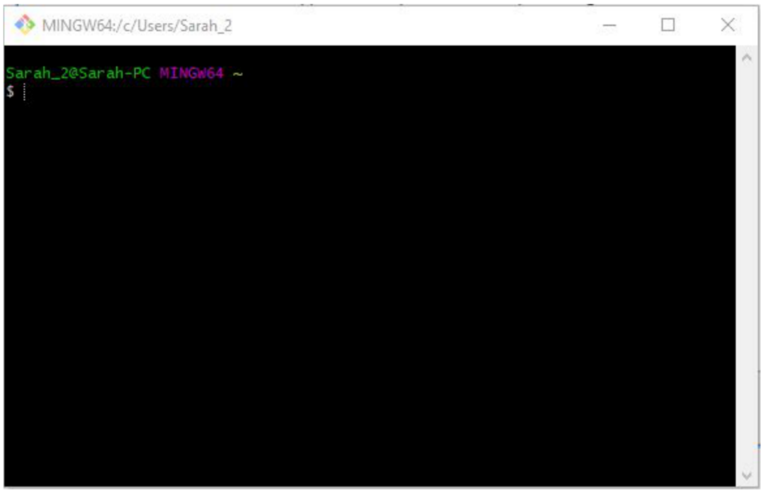
Click Finish to exit Setup.

☑ Launch Git Bash

□ View Release Notes

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.



```
Sarah_2@Sarah-PC MINGW64 ~
$ 
```

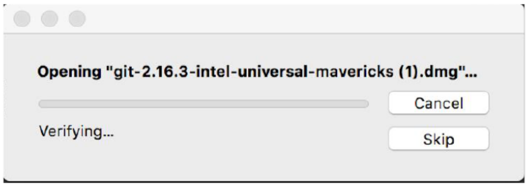
Git Bash is the command line interface you will use to configure Git

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.

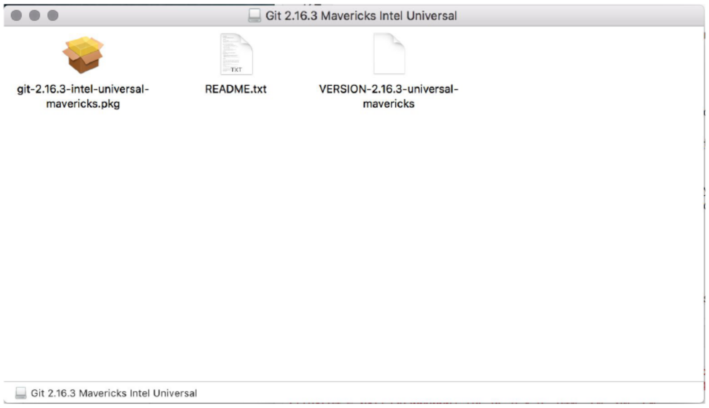
After downloading the appropriate Git version for Macs, you should have downloaded a DMG file for installation on your Mac. Open this file. This will install Git on your computer. A new window will open.

Macs installation process



Opening "git-2.16.3-intel-universal-mavericks (1).dmg"...

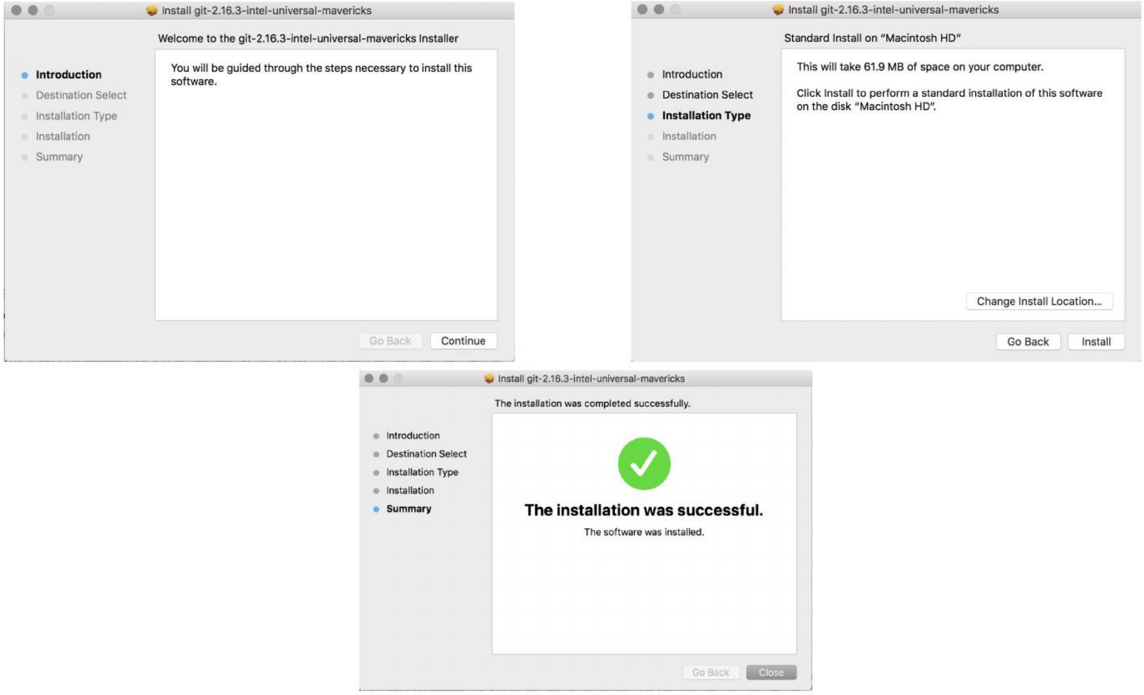
Verifying...



Git 2.16.3 Mavericks Intel Universal
git-2.16.3-intel-universal-mavericks.pkg
README.txt
VERSION-2.16.3-universal-mavericks
Git 2.16.3 Mavericks Intel Universal

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!



Install git-2.16.3-intel-universal-mavericks

Welcome to the git-2.16.3-intel-universal-mavericks Installer

You will be guided through the steps necessary to install this software.

Install git-2.16.3-intel-universal-mavericks

Standard Install on "Macintosh HD"

This will take 61.9 MB of space on your computer.

Click Install to perform a standard installation of this software on the disk "Macintosh HD".

Install git-2.16.3-intel-universal-mavericks

The installation was completed successfully.

The installation was successful.

The software was installed.

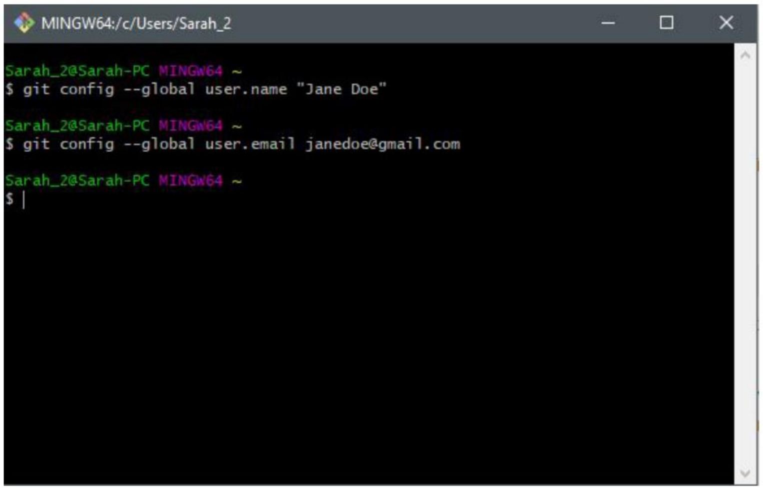
Steps to a successful installation of Git!

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!**

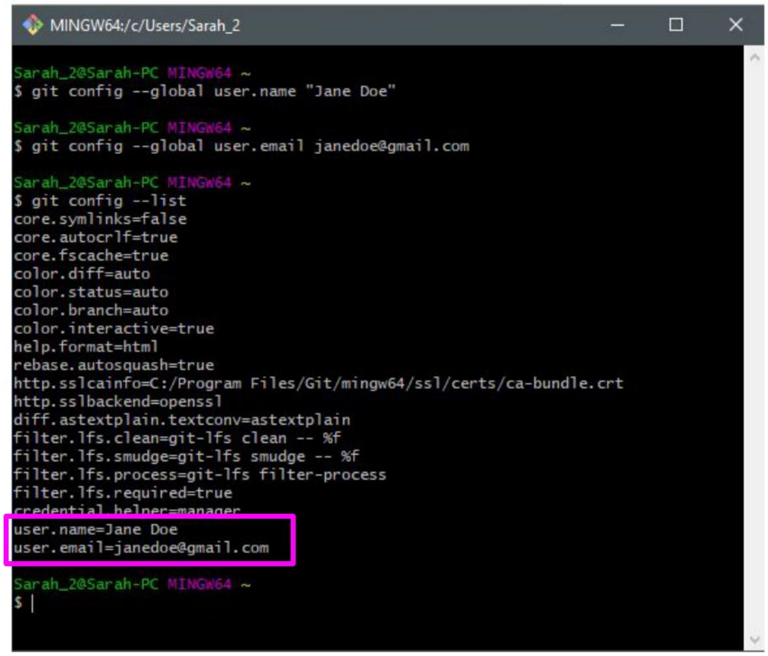


```
Sarah_2@Sarah-PC MINGW64 ~
$ git config --global user.name "Jane Doe"

Sarah_2@Sarah-PC MINGW64 ~
$ git config --global user.email janedoe@gmail.com

Sarah_2@Sarah-PC MINGW64 ~
$ |
```

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`

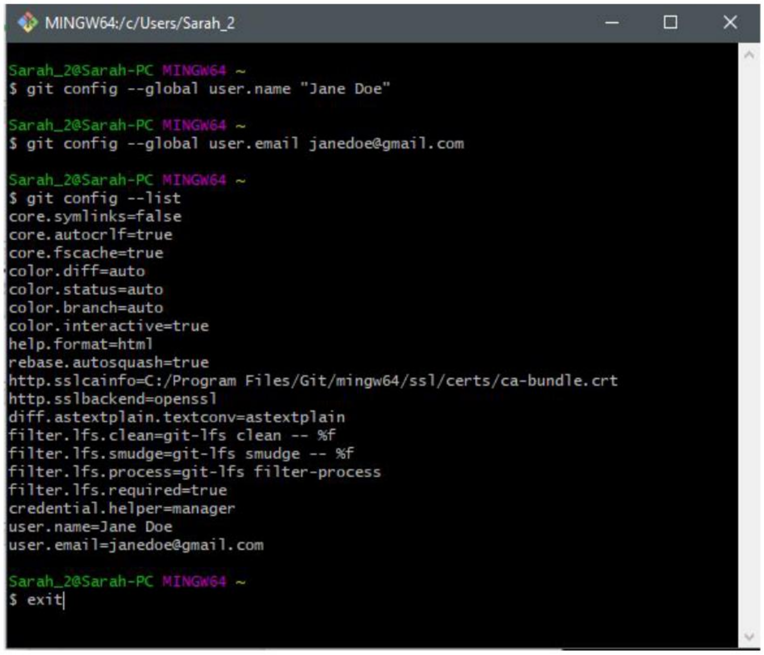


```
Sarah_2@Sarah-PC MINGW64 ~
$ git config --global user.name "Jane Doe"

Sarah_2@Sarah-PC MINGW64 ~
$ git config --global user.email janedoe@gmail.com

Sarah_2@Sarah-PC MINGW64 ~
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
rebase.autosquash=true
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
http.sslbackend=openssl
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
credential.helper=manager
user.name=Jane Doe
user.email=janedoe@gmail.com

Sarah_2@Sarah-PC MINGW64 ~
$ |
```

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!



```
Sarah_2@Sarah-PC MINGW64 ~
$ git config --global user.name "Jane Doe"

Sarah_2@Sarah-PC MINGW64 ~
$ git config --global user.email janedoe@gmail.com

Sarah_2@Sarah-PC MINGW64 ~
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
rebase.autosquash=true
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
http.sslbackend=openssl
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
credential.helper=manager
user.name=Jane Doe
user.email=janedoe@gmail.com

Sarah_2@Sarah-PC MINGW64 ~
$ exit
```

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.