Creating and Hosting a Personal Website on Github

The Plan

I. Git, GitHub, and GitHub Pages

II. Building your personal website.

III. What is Jekyll?

IV. Bonus: Getting fancy with Jekyll

Git, GitHub, and GitHub Pages

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.

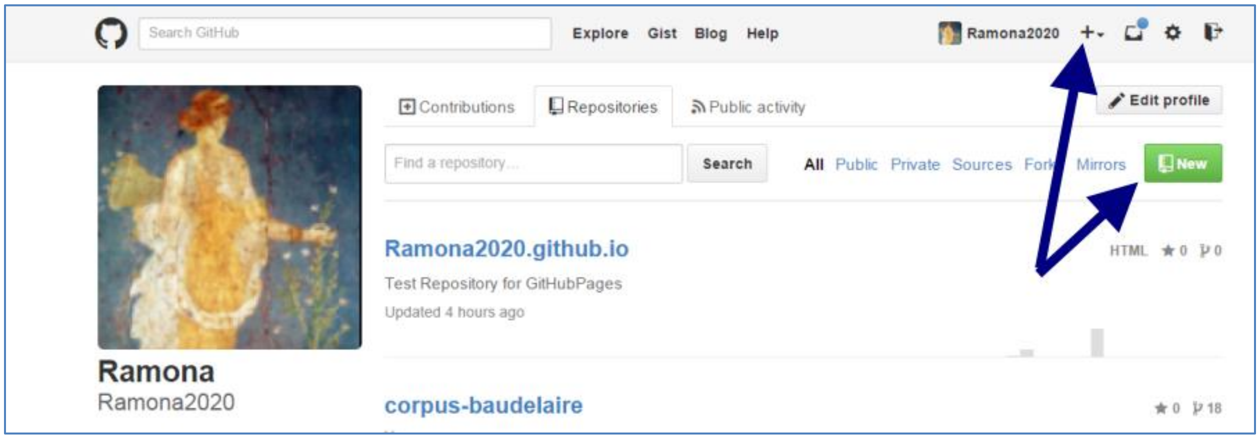
GitHub, http://github.com, is a web-based Git repository hosting service with all of the revision control and source code management of Git plus some added features. GitHub provides a web-based graphical interface and desktop as well as mobile integration.

GitHub Pages, https://pages.github.com/ are public webpages hosted for free through GitHub. GitHub users can create and host both personal websites (one site per user account) and websites related to specific GitHub projects.

Building Your Personal Website

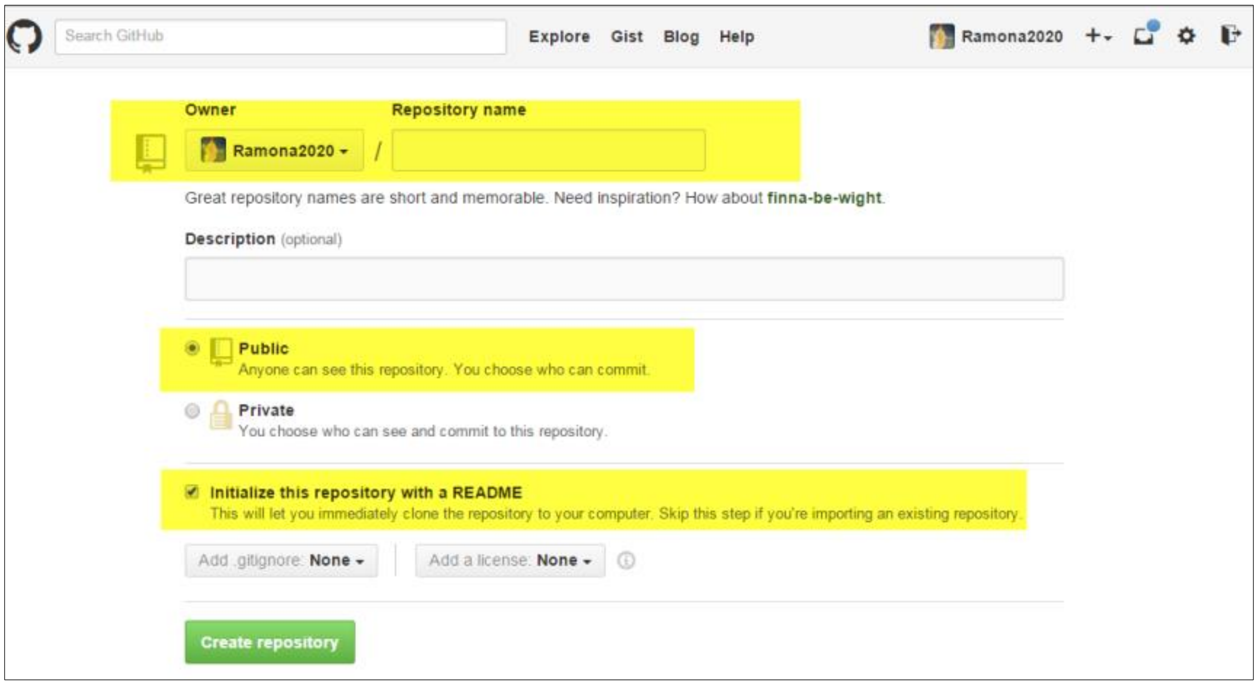
Creating Your Website

1. Login to your GitHub account and click on the **New** repository icon from your account homepage or the plus sign in the top right corner.



2. Name your repository *username*.github.io, replacing *username* with your GitHub username. For example, my username is Ramona2020 and so my repository would be

Ramona2020.github.io
2.1 Make sure the repository is set to public
2.2 Check the box to Initialize this repository with a README file



Owner Ramona2020 / Repository name

Great repository names are short and memorable. Need inspiration? How about **finna-be-wight**.

Description (optional)

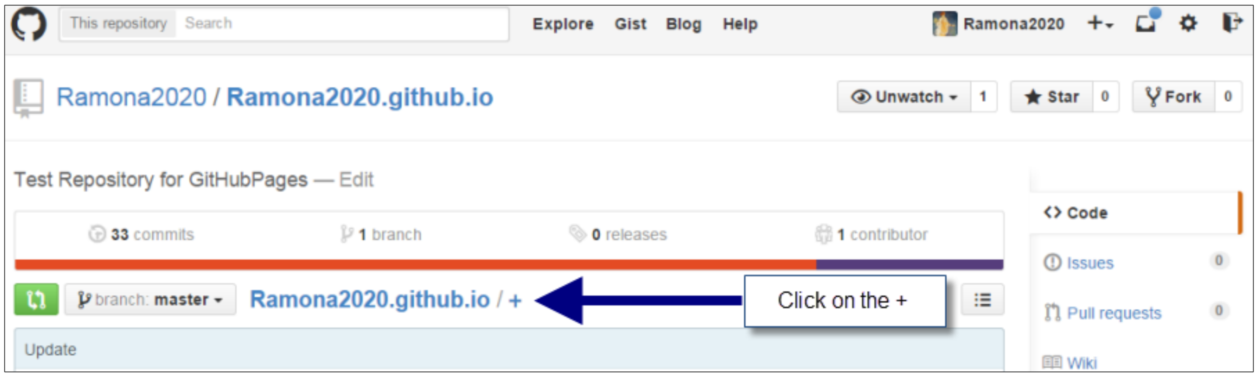
● **Public**
Anyone can see this repository. You choose who can commit.

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

☑ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** Add a license: **None**

3. Create an **index.html** page by clicking the plus icon next to your repository name and typing the file name directly into the input box that appears.



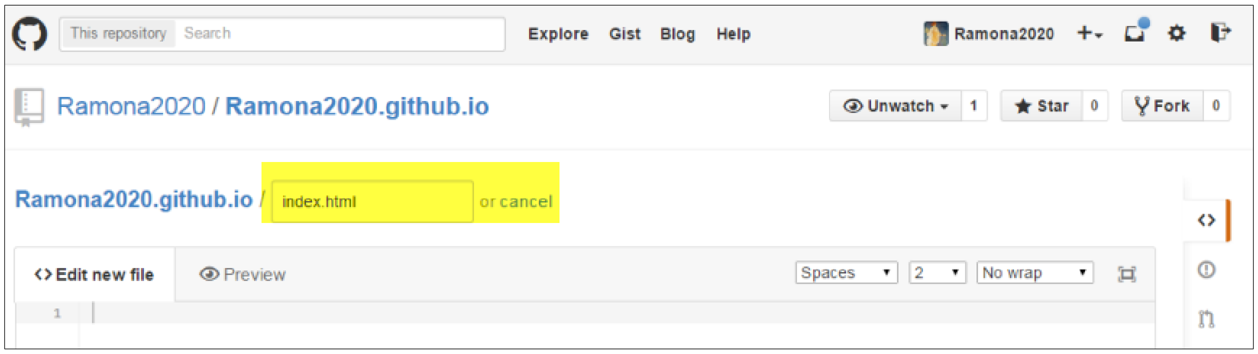
Ramona2020 / **Ramona2020.github.io**

Test Repository for GitHubPages — Edit

33 commits **1** branch **0** releases **1** contributor

branch: **master** Ramona2020.github.io / +

Update



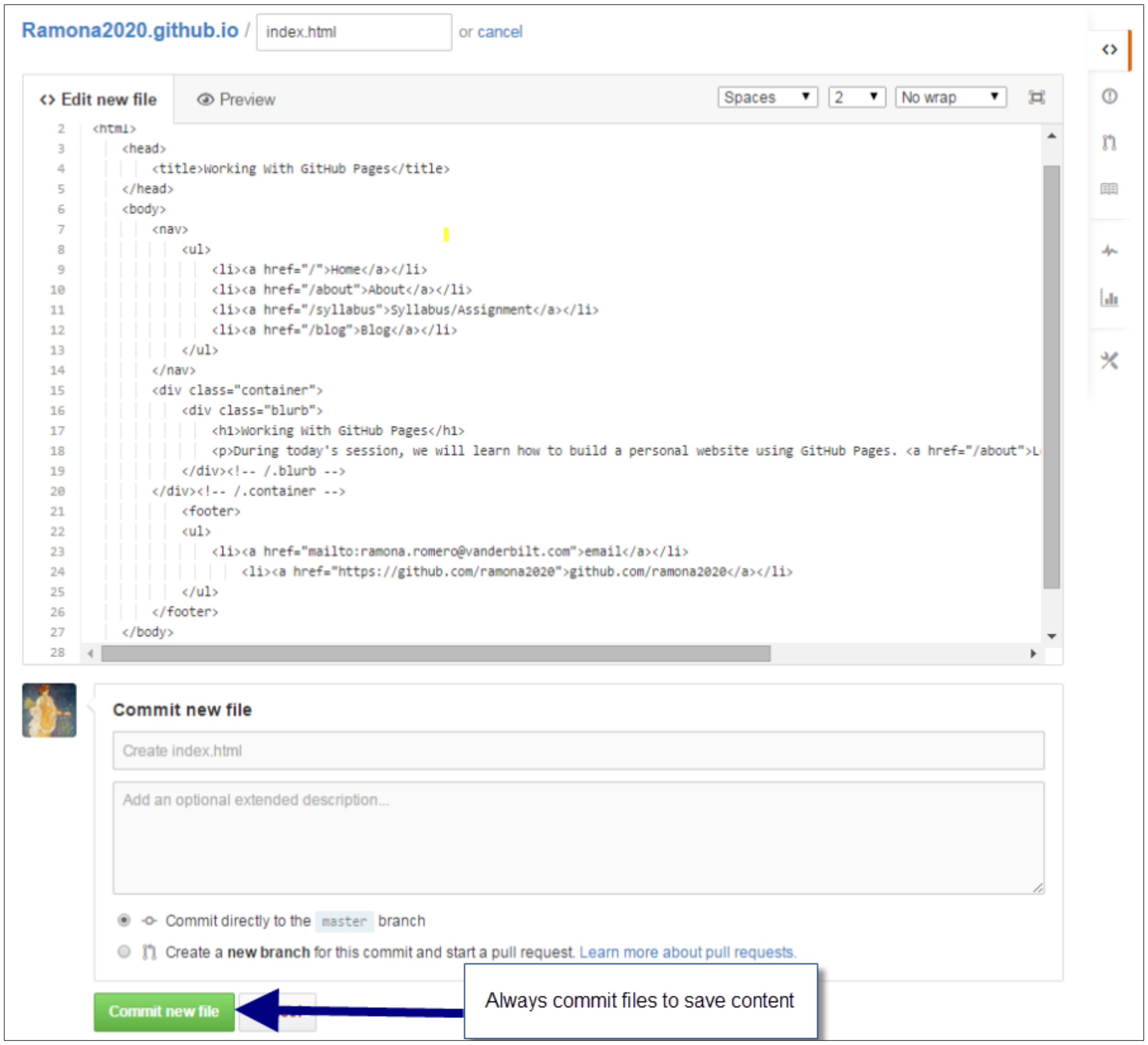
Ramona2020 / **Ramona2020.github.io**

Ramona2020.github.io / index.html or cancel

4. A text editor will open. Insert the following html code:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Working With GitHub Pages</title>
    </head>
    <body>
        <nav>
            <ul>
                <li><a href="/">Home</a></li>
                <li><a href="/about">About</a></li>
                <li><a href="/syllabus">Syllabus/Assignment</a></li>
                <li><a href="/blog">Blog</a></li>
            </ul>
        </nav>
        <div class="container">
            <div class="blurb">
                <h1>Working With GitHub Pages</h1>
                <p>During today's session, we will learn how to build a
personal website using GitHub Pages. <a href="/about">Learn about Git,
GitHub, and GitHub Pages</a></p>
            </div><!-- /.blurb -->
        </div><!-- /.container -->
            <footer>
            <ul>
                <li><a
href="mailto:ramona.romero@vanderbilt.com">email</a></li>
                    <li><a
href="https://github.com/ramona2020">github.com/ramona2020</a></li>
            </ul>
        </footer>
    </body>
</html>
```

In order to save the changes you've just made you must **Commit** your file at the bottom of the page.


Ramona2020.github.io / index.html or cancel
<> Edit new file
Preview
Spaces
2
No wrap

```
<html>
  <head>
    <title>Working With GitHub Pages</title>
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="/">Home</a></li>
        <li><a href="/about">About</a></li>
        <li><a href="/syllabus">Syllabus/Assignment</a></li>
        <li><a href="/blog">Blog</a></li>
      </ul>
    </nav>
    <div class="container">
      <div class="blurb">
        <h1>Working With GitHub Pages</h1>
        <p>During today's session, we will learn how to build a personal website using GitHub Pages. <a href="/about">L
      </div><!-- /.blurb -->
    </div><!-- /.container -->
      <footer>
      <ul>
        <li><a href="mailto:ramona.romero@vanderbilt.com">email</a></li>
          <li><a href="https://github.com/ramona2020">github.com/ramona2020</a></li>
      </ul>
    </footer>
  </body>
```

Commit new file
Create index.html
Add an optional extended description...
Commit directly to the master branch
Create a new branch for this commit and start a pull request. Learn more about pull requests.
Commit new file
Always commit files to save content

This will create your GitHub Pages Site. You can view your page at http://*username*.github.io.

Styling Your Content

1. Style your content by creating a css file. To create the css file go back to your repository home and click on the plus sign next to your repository name. Type in **css/main.css**.
   1.1 Typing **css/** before the file name creates a subdirectory called **css** that will contain your CSS file.
   1.2 A text editor will open. Insert the following code:

```
body {
    margin: 60px auto;
    width: 70%;
}
nav ul, footer ul {
    font-family:'Helvetica', 'Arial', 'Sans-Serif';
    padding: 0px;
    list-style: none;
    font-weight: bold;
}
nav ul li, footer ul li {
    display: inline;
    margin-right: 20px;
}
a {
    text-decoration: none;
    color: #999;
}
a:hover {
    text-decoration: underline;
}
h1 {
    font-size: 3em;
    font-family:'Helvetica', 'Arial', 'Sans-Serif';
}
p {
    font-size: 1.5em;
    line-height: 1.4em;
    color: #333;
}
footer {
    border-top: 1px solid #d5d5d5;
    font-size: .8em;
}

ul.posts {
    margin: 20px auto 40px;
    font-size: 1.5em;
}

ul.posts li {
    list-style: none;
}
```

Remember to commit your file to save changes!

Link your CSS file inside your HTML document's head. Go back to the index.html file that you created and click on the pencil icon to edit this file. Add the following code highlighted in green to the head element in your HTML document.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Working With GitHub Pages</title>
        <!-- link to main stylesheet -->
        <link rel="stylesheet" type="text/css" href="/css/main.css">
    </head>
    <body>
```

Congratulations! You now have the framework for a basic website that you can customize to suit your needs.

What is Jekyll?

Jekyll, http://jekyllrb.com/, is a simple, blog-aware, static site generator. It takes a template directory containing raw text files in various formats, runs it through a converter (like Markdown) and Jekyll's Liquid renderer, and spits out a complete, ready-to-publish static website suitable for serving with your favorite web server. Jekyll also happens to be the engine behind GitHub Pages, which means you can use Jekyll to host your project’s page, blog, or website from GitHub’s servers for free.

Examples of sites using Jekyll:

Short list: http://jekyllrb.com/docs/sites/

Long list: https://github.com/jekyll/jekyll/wiki/Sites

Bonus: Getting fancy with Jekyll

The following steps will allow you to take advantage of some of Jekyll's features in your current website.

Create a .gitignore file

Creating a .gitignore file tells Git to ignore the _site directory that Jekyll automatically generates each time you commit. Since this directory and all the files that it contains are written every time you commit a change, there's no need for version control.

1. Go to your repository home and and click on the plus sign next to your repository name and type in **.gitignore**

2. In the text editor, type the following: **_site/**

3. Commit your .gitignore file

Create a _config.yml file

Creating a **_config.yml** file tells Jekylly some basics about your project.

1. Go to your repository home and and click on the plus sign next to your repository name and type in **_config.yml**

2. in the text editor, type the following and commit your file:

```
name: Working with GitHub Pages
markdown: redcarpet
pygments: true
authors:
  Ramona:
name: Ramona Romero
display_name: Ramona
email: ramona.romero@vanderbilt.edu
twitter: Ramona2020
github: Ramona2020
```

Edit the above to incorporate your name and information in the authors section as well as the name of your site.

Create a _layouts directory and default.html file

The **default.html** is our main layout for the website and will contain repeated elements like the`<head>` and `<footer>`. It's like creating a shortcut for incorporating repeatable elements.

1. Go to your repository home and click on the plus sign next to your repository name. Type in **_layouts/default.html.**
   1.1 Typing **_layouts** before the file name creates a subdirectory called _layouts that will contain your **default.hmtl** file.
   1.2 A text editor will open. Insert the following code:

```
<!DOCTYPE html>
    <html>
        <head>
            <title>{{ page.title }}</title>
            <!-- link to main stylesheet -->
            <link rel="stylesheet" type="text/css" href="/css/main.css">
        </head>
        <body>
            <nav>
                <ul>
                    <li><a href="/">Home</a></li>
                    <li><a href="/about">About</a></li>
                    <li><a href="/syllabus">Syllabus/Assignments</a></li>
                    <li><a href="/blog">Blog</a></li>
                </ul>
            </nav>
            <div class="container">

            {{ content }}

            </div><!-- /.container -->
            <footer>
                <ul>
                    <li><a
href="mailto:ramona.romero@vanderbilt.com">email</a></li>
                    <li><a
href="https://github.com/ramona2020">github.com/ramona2020</a></li>
                </ul>
            </footer>
        </body>
    </html>
```

Update Your index.html file with Your New Default Layout

1. Go to your repository home and click on the index.html link.

2. To edit this file click on the pencil icon in the toolbar above your code to the right of your screen.

3. Paste the following code in the text editor and commit the file:

```
---
layout: default
title: Working With GitHub Pages
---

    <div class="container">
        <div class="blurb">
            <h1>Working With GitHub Pages</h1>
            <p>During today's session, we will learn how to build a personal
website using GitHub Pages. <a href="/about">Learn about Git, GitHub, and
GitHub Pages</a></p>
        </div><!-- /.blurb -->
    </div><!-- /.container -->
```

Note in the above code that we no longer need to include information about the header, footer or page navigation as all of this information is contained in the default.html file in the _layouts folder.

Creating a Blog

We can create a Jekyll-based blog following the same conventions we just used to create our default layout.

Creating a default layout for posts

Just as we created a default look and feel for our website, we are going to do the same for our blog posts.

1. Go to your repository home and click on **_layouts** folder
2. Click on the plus sign near your repository name to create a new file called **post.html**.
3. Enter the following code into the text editor and commit the file:

```
---
layout: default
---

<h1>{{ page.title }}</h1>
<p class="meta">{{ page.date | date_to_string }}</p>

<div class="post">
    {{ content }}
</div>
```

Creating a _posts directory and your first blog post.

All of the blog posts that you create will live in the **_posts** directory. Jekyll has very strict naming conventions for files. Your posts must follow the following naming convention: **YYYY-MM-DD-title-of-my-post.md.**
Please note that the file extension .md stands for Markdown, a markup language that's easy to learn and use for blog authoring. Learn more at http://daringfireball.net/projects/markdown/.

1. Go to your repository home and click on the plus sign near your repository name.

2. Type the following to create the _posts directory and your first blog post: **_posts/2015-04-03-my-post.md**

3. In the text editor insert the following code and commit your file:

```
---
layout: post
title: My First Post
date: 2015-04-03
---

Write something about your experience in this workshop.
```

Creating the blog directory on your website

You've created your first blog post, but now we need to make it accessible on your website.

1. Go to your repository home and click on the plus sign near your repository name.

2. Type the following to create the blog directory and the index.html file that will live inside of it: **blog/index.html**

3. In text editor type the following code and commit the file:

```
---
layout: default
title: GitHub Pages Blog
---
<h1>{{ page.title }}</h1>
<ul class="posts">

  {% for post in site.posts %}
    <li><span>{{ post.date | date_to_string }}</span> &raquo; <a href="{{
post.url }}" title="{{ post.title }}">{{ post.title }}</a></li>
  {% endfor %}

</ul>
```

You should now be able to navigate to the blog page of your website and see your first post listed and linked there.

Customize your blog post URLs

You can easily customize your blog post URLs by editing the **_config.yml** file that created earlier. 1. Go to your repository home and click on **_config.yml** link.

2. To edit this file click on the pencil icon in the toolbar above your code to the right of your screen.

3. Add the following link at the end of your file and commit it:

```
permalink: /blog/:year/:month/:day/:title
```

Adding an RSS Feed to your blog

1. Go to your repository home and click on **blog** folder
2. Click on the plus sign near your repository name to create a new file called **atom.xml**.
3. Enter the following code into the text editor and commit the file:

```
---
layout: feed
---
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">

 <title>GitHub Pages Blog</title>
 <link href="http://ramona2020.github.io/blog/atom.xml" rel="self"/>
 <link href="http://ramona2020.github.io/blog"/>
 <updated>{{ site.time | date_to_xmlschema }}</updated>
 <id>http://ramona2020.github.io/blog</id>
 <author>
   <name>Ramona Romero</name>
   <email>ramona.romero@vanderbilt.edu</email>
 </author>

 {% for post in site.posts %}
 <entry>
   <title>{{ post.title }}</title>
   <link href="http://ramona2020.github.io{{ post.url }}"/>
   <updated>{{ post.date | date_to_xmlschema }}</updated>
   <id>http://ramona2020.github.io/{{ post.id }}</id>
   <content type="html">{{ post.content | xml_escape }}</content>
 </entry>
 {% endfor %}

</feed>
```

Wrapping Up

Congratulations! You now have the basic framework of your website up and running. Now you can easily customize your content.

All resources for today's workshop (handouts, web links…) are available online at;

https://github.com/HeardLibrary/workshops/tree/master/GithubPages