**Setting up GitHub Pages via Gem, Bundler and Jekyll:**

* Prerequisites:
  + Before you can use Jekyll to create a GitHub Pages site, you must install Git and Jekyll:
    - For Git, see here: <https://help.github.com/en/github/getting-started-with-github/set-up-git>
    - Make sure you have Git Bash installed as well;
    - You can follow the instructions at <https://programminghistorian.org/en/lessons/building-static-sites-with-jekyll-github-pages>
    - For Jekyll, it is recommended to use *Bundler* to install and run it. You will need to install *Ruby and Bundler*:
      * For *Ruby*, follow the instructions here: <https://www.ruby-lang.org/en/documentation/installation/>
      * For Windows, I chose the *RubyInstaller* and then chose *Ruby+Devkit 2.6.6-1(x64)* as recommended;
      * Once Ruby has installed correctly, you should be able to install *Bundler* by following the instructions at this link: <https://bundler.io/>
* Next step is to create a repository for your site:
  + You can either create a new one or create the site in an existing repository
  + If you want to create a new repository, follow the instructions on this link: <https://help.github.com/en/github/working-with-github-pages/creating-a-github-pages-site-with-jekyll#creating-a-repository-for-your-site>
  + If you want to use an existing repository, you can skip forward to creating your site: <https://help.github.com/en/github/working-with-github-pages/creating-a-github-pages-site-with-jekyll#creating-your-site>
  + In Git Bash, navigate to the folder your repository is in, and follow the instructions:
    - Input ***gem install Jekyll bundler*** to install Jekyll and bundler
    - Input ***jekyll new website*** where ***website*** is the name of the website folder where all the necessary files will be installed to.
    - Next, navigate to the folder using ***cd website***
    - If you search for the folder on File Explorer, you will see that a bunch of new files (files that will run your website) have been installed!
* To run the website locally (meaning you can experiment on it all without making changes publicly) enter:
  + ***bundle exec jekyll serve - - watch***
  + While site is running and after making changes to website files: save the files and refresh the webpage to see the changes (**except for the \_config.yml file**, for which you need to stop the website running, and restart to see the changes);
  + To stop running the site locally, press ***control-c***
  + To view the locally running site, visit **localhost:4000** to see the basic Jekyll website with boilerplate text;
* **Basic site settings via \_config.yml:**
  + Navigate to the website directory folder to the \_config.yml files which is the main settings file – open in your preferred text editor;
  + Edit the \_config.yml file according to these instructions:
    - **title:** The title of your website, as you want it to appear in the header of the webpage.
    - **email:** Your email address.
    - **description:** A description of your website that will be used in search engine results and the site’s RSS feed.
    - **baseurl:** Fill in the quotation marks with a forward slash followed by the name of your website folder (e.g. “/website”) to help locate the site at the correct URL.
    - **url:** Replace “http://yourdomain.com” with “localhost:4000” to help locate your local version of the site at the correct URL.
    - **twitter\_username:** Your Twitter username (do not include @ symbol).
    - **github\_username:** Your GitHub username.
  + The changes you made to the baseurl and url lines will let your site run from the same files both locally on your computer and live on the Web, but doing this changed the URL where you’ll see your local site from now on (while Jekyll is running) from localhost:4000 to localhost:4000/website/ (substitute your website folder name for website and remembering the last slash mark).

**Where (and what) is everything?**

* **\_config.yml** is discussed above; it provides basic settings information about your site, such as the site’s title and additional possibilities we won’t cover here, like how to structure links to posts (e.g. should they follow the pattern MySite.com/year/month/day/post-title?).
* **\_includes folder** has files that get included on all or certain pages (e.g. code to make the header contain your site title and main menu on every page of the site)
* **\_layouts folder** contains code that controls how the pages on your site look (default.html), as well as customizations of that code to further style blog posts (post.html) and pages (page.html)
* **\_posts folder** holds the individual files that each represent a blog post on your website. Adding a new post to this folder will make a new blog post appear on your website, in reverse chronological order (newest post to oldest). We’ll cover adding blog posts in the next section.
* **\_sass folder** holds SCSS files that control the visual design of the site
* **\_site folder** is where the HTML pages that appear on the web are generated and stored (e.g. you’ll write and save posts as Markdown files, but Jekyll will convert these to HTML for display in a web browser)
* **index.md** is a place to add content that you want to appear on your homepage, such as a biography blurb to appear above the “Posts” list
* **about.md** is an example of a Jekyll page. It’s already linked in the header of your website, and you can customize its text by opening and writing in that file. We’ll cover adding more site pages in the next section.
* **css folder** holds CSS converted from SCSS that controls the visual design of the site
* **feed.xml** lets people follow the RSS feed of your blog posts
* **index.html** controls the structuring of content on your site’s homepage