How to Use Jekyll

Carolyn Anderson
last updated May 20, 2015

Jekyll is a program for managing websites. Jekyll is a **static site generato**r. It allows you to write templates and apply them to html code to **generate websites**. Currently, the Learn Mi'gmaq project is using only a few features of Jekyll: **template wrapping**, **page variables**, and **code includes**.

To generate a website using Jekyll, you can write the webpage just as you normally would, using HTML, CSS, and Javascript. Then, you can use a **markup language** to tell Jekyll how to add extra pieces of code. When you want to update the published website, Jekyll takes the webpage, runs it through the layout files specified by your markup instructions, and generates a ready-to-upload webpage.

At its simplest, Jekyll is glorified cut-and-paste, allowing you to modify multiple webpages at the same time. It also has more complex features that we aren't using.

Getting Started

By far the trickiest part of Jekyll is simply installing it. Jekyll is not available for Windows. You will need to be running Linux or Mac OS in order to use Jekyll.

1. First, make sure you have the requirements for Jekyll. This includes **Ruby** and **RubyGems**. The easiest way to install these is using the command line. You may need to reinstall Ruby to make sure your copy is up-to-date: you may also need to run **sudo** apt-get install instead of just apt-get install.

2. Once you have RubyGems installed, you can install Jekyll by running *gem install jekyll* on the command line.

3. That's all that is needed to install Jekyll itself. Next, you need to figure out where to run Jekyll, that is, what your **jekyll directory** is. For the Learn Mi'gmaq project, this is whatever you have named your copy of the Bitbucket repository.

Jekyll Directory Structure

Jekyll applies different sets of instructions to your files based on where they are stored. All Jekyll directories have the following locations:

main:

This is the main directory. Any .html files in this folder that have a YAML Front Matter section (more on that later) will be transformed by Jekyll. You can also make new folders and Jekyll will transform .html files in those as well.

jekyll/_site:

This is the destination folder. All files that Jekyll transforms will be placed here. This folder is cleared every time Jekyll is run. **Do not store files in here. They will be deleted!** This is the location where your final webpages will be produced.

jekyll/_layouts:

This is where the layouts that Jekyll applies to your webpages are stored. Layouts are also .html files, but with special markups to indicate formatting. More on writing layouts to come.

jekyll/_includes

This folder contains code snippets that you are including in your files. Jekyll's markup language allows you to insert the names of code snippet files to be included. This allows you to modify pieces of code that appear on many webpages in just one place. The code snippets are also .html files.

jekyll/css

Here is where all your CSS stylesheets belong. Jekyll will copy this directory wholesale into the *_site* directory each time the website is generated.

jekyll/img

Here is where all images are stored. Jekyll copies this directory wholesale into the *_site* directory each time the website is generated.

jekyll/is

All Javascript files can be found here. Jekyll copies this directory wholesale into the *_site* directory each time the website is generated.

Running Jekyll

To just run Jekyll and output webfiles, this is what you do:

1. Enter the *jekyll* directory.
2. Run the command: *jekyll build*
3. The website will be generated in the *_site* folder.

Makefile

For the Learn Mi'gmaq folder, I have set up a Makefile. This is a file that runs a bunch of commands automatically for you, so that you don't have to type a lot of different commands in.

All you need to type is:

make

in the terminal, and the Make program will run scripts to update any files if you have made modifications.

So if you want to make sure everything is up-to-date, just run *make* instead of *jekyll build*.

Adding a New Webpage to Jekyll

1. First, copy the webpage into the Jekyll directory.

2. Next, add a layout markup to the top.

3. The beginning of a markup is indicated by three dashes on a line: --- (no spaces)

4. Next, on a new line, write *layout: myLayout.html*, where myLayout.html is the file name of the layout.

5. Then, end the markup with another line of three dashes.

6. Your website is now ready to be transformed by Jekyll. But you may need to write your layout file first.

Layouts

Currently there are two layouts used on the Learn Mi'gmaq site: frame.html and lesson.html. Frame.html is for non-lesson pages; lesson.html is for sections, units, and lessons. Frame.html should serve most of your new-page needs, but if you want to do soemthing special, you might also need to write your own layout.

Writing a New Layout File

1. A layout is essentially a template that gets wrapped around your content. Jekyll was designed for blog management, so a classic example of a layout is a header and footer that are wrapped around each blog post.

2. To write a new layout, enter the *_layouts* directory.

3. Create a new html file for your layout.

4. The essential component of a layout is the markup to include the content. This is the line *{{content}}*. It should be placed wherever the html code that differs on each page appears.

5. Fixed content (code that is shared by all pages using this template) can be placed anywhere around this line. This content can be normal html code.

Including Code Snippets

One very useful feature of Jekyll is that it allows you to include code snippets in your webpage. Without this feature, if you had a piece of code that appeared on multiple webpages, you would have to update them all separately each time you edited that code. Code includes allows you to put the code in one html file in the *_includes* directory and edit it there. Jekyll automatically pastes the code into the files you have included it in each time it generates the website.

1. Create a new html file in the *_includes* directory.

2. Paste the code that you want to include into the file.

3. At each place in a webpage where you want to use this code, write the line *{% include fileName.html %}*, where fileName.html is the name of your include file.

Using Variables

The markup at the beginning of each webpage can also be used to define variables used in your code. For instance, the standard layout used by the Learn Mi'gmaq project sets the title of the webpage to a variable called *title*:

```
<title>{{ page.title }}</title>
```

This variable is prefixed with page because it is defined in the markup for each webpage, as follows:

```
- - -
layout: layoutName.html
title: myPageTitle
- - -
```

This sets the title of each webpage to be whatever is specified in the markup at the beginning of each webpage.