What Icon Fonts are available for J1 Template?

J1 Template supports popular icons font sets out-of-the-box. The icon sets have a rich set of different icons for many categories used for the Web. Using J1 Template, the following icons font are available out-of-the-box:

- Material Design Icons
- Material Design Light Icons
- FontAwesome Icons V5
- Iconify Icons

MDI and FA icons are very good in design and have a rich set of different icons for many categories used for the Web.

Find more details for Material Design Icons and FontAwesome Icons on http://localhost:41000/pages/public/learn/roundtrip/mdi_icon_font/.

What are Asciidoc Paragraphs?

The primary block type in most documents is the paragraph. That's why in Asciidoc, you don't need to use any special markup or attributes to create paragraphs. You can just start typing sentences and that content becomes a paragraph.

This page introduces you to the paragraph in AsciiDoc and explains how to set it apart from other paragraphs.

Find more on https://docs.asciidoctor.org/asciidoc/latest/blocks/paragraphs/

What is Rocketstart?

Using Rocketstart, no software needs to be installed locally on your computer to create J1 Template based website. Create a site on the Internet in minutes.

The people on Netlify provide a really easy workflow to bring a website to life using a template like JekyllOne Template on the Internet in minutes: The 1-Click Deployment.

The Workflow allows users to deploy a website on the Internet without a locally running installation on their computers. I invite you on a journey to using the 1-Click Deployment for your new site on the Internet.

Two service providers are required running an 1-Click Deployment: Github and Netlify. The platforms provided by Github and Netlify are used to build and control the J1 website on the Internet. Both providers give full access to their services with no limits.

Free services with no restrictions are not the case with all digital service providers. Therefore, I can recommend these two service providers for entry into the world of modern digital services.

If you do not already have accounts, sign up to be prepared for the journey:

- Sign up on Github
- Sign up on Netlify

It is recommended to sign up for Github first. You can use the account data from Github to register on Netlify. At the end of this journey, which only takes a few minutes, you will have your presence, a website on the Internet.

You don't need any software to be installed on your computer. You won't pay anything for this trip. Your new website on the Internet will also be free. Find more on https://jekyll.one/pages/public/learn/rocketstart/.

What is a Static Site Generator?

Static website generators (SSGs) are programs that create HTML pages of a website from text-based template files or already prepared HTML sources. A SSG represent an alternative to a database-based system (CMS) such as WordPress. In content management systems, the content of a website is not managed in simple text files but stored in (relational) databases such as MySQL.

A typical CMS-driven website works by building each and every page dynamically. That means fetching the content from a database to pass them over to a template system (engine) to load the page content, adding the required structure information (e.g., CSS styles) and Javascript components if needed. This means every page is assembled on demand.

A Static Site Generator takes a different approach and generates all the pages of the website upfront. That does not necessarily mean that a static website is always built from scratch. Modern generators support a so-called incremental build, as does. When there are changes detected to the content, only selected pages will rebuild.

In the case of blogs, documentation sites, or company websites, CMS and website builders are used for this, but static site generators have distinct advantages over a typically much more complex content management solution:

- Speed: Websites created with a static website generator are characterized by excellent user speed. Files are processed when the page is created and not when the user views the page.
- Version Control: Although the content of dynamic web projects is stored separately from the code in databases, a static website usually resides in simple text files that are fairly easy to maintain. Structurally, the content elements do not differ from other parts of the code base, so version management is set up without any problems.
- Security: The advantage of websites built with static website builders is that they offer little potential for attack, unlike, for example, content management systems like WordPress, which are vulnerable to security vulnerabilities and need to be updated regularly. The risk potential is limited to the one-off access by the customer when accessing the site. Since this is usually not about more than the delivery of structured HTML pages, the probability of unwanted access is minimal.
- Easy Maintenance: The number of components required for SSGs is comparatively small, but these are only relevant during development. While other solutions for live operation require various modules, databases, libraries, frameworks, and packages and have to be updated regularly, static pages are only tied to a functioning web server.

Find more to know:

- J1 Theme at Jamstack Club
- Popular SSGs and Themes at Jamstack Club
- Popular SSGs and Themes at Jekyll Themes

What is Jekyll?

Jekyll is a popular Static Site Generator (SSG). As the name SSG implies, the generator component creates websites, but Jekyll is not a content management system (CMS) like WordPress. A static site generator is a perfect toolset to create and manage personal, project, or documentation sites because they do not rapidly change their content.

The generator concept dispenses with the convenience of a website editing system using centralized stored content in databases, in favour of editing simple files with the help of text editors. SSGs like Jekyll read the code of a content page from simple text files that contain rendering information already.

The makers of Jekyll describe it this way: Think of it like a file-based CMS, without all the complexity. Jekyll takes your content, renders Markdown and Liquid templates, and spits out a complete, static website ready to be served by Apache, Nginx or another web server.

Find more to know:

- The Philosophy of Jekyll

How to install Jekyll?

To install and run Jekyll, the dynamic programming language Ruby is required. Jekyll is a Ruby Gem that is supported on all current Windows, Linux and MacOS systems.

Run the command gem below in a terminal to install Jekyll: gem install Jekyll

Find more at RubyGems to learn what Ruby Gems are and how to use and manage them. The builder engine Jekyll is supported by all current Ruby versions v2.7 or higher. To achieve the best performance for building websites, the most recent version V3 of Ruby is recommended.

The Ruby Gem of Jekyll contains a command line interface (CLI) and a library system to extended and customize the builder engine.

Find more to know:

- Install Jekyll
- Ruby 101
- Jekyll 101

How to use Jekyll?

To use Jekyll, it is needed to learn the basics of the static websites' development. The builder engine is a command line interface (CLI) and no Graphical User Interface (GUI) or Web Interface (UI) managed by a web browser is provided.

All websites created by Jekyll are based on Themes, a template system to ease the creation process of web pages, design, styles, navigation elements and more. For a wide range of website types, the J1 Theme accessible at the Jamstack Club can be used.

Check the preview function and, if the Jekyll Theme meet your needs, you can create a example web from here: Rocketstart, Create A Internet Site In Minutes Start learning the development of static webs using Jekyll the easy way.

Find more to know:

- Jekyll 101
- Jekyll in a Day

Is Jekyll still used?

The latest version of Jekyll v4.3.1 is issued in October 2022 and version v4.4.0 is being prepared for 2023. The next version of Jekyll V5 is under construction and a tentative roadmap is discussed at Github: Jekyll v5 Roadmap.

Find more to know:

- Jekyll News

Is Jekyll a CMS?

Jekyll is a static site generator (SSG), not a content management system (CMS) in the traditional sense.

A CMS is a software application that allows users to create, manage, and publish digital content on the web, often providing features such as content creation, editing, media management, user management, and more. Examples of popular CMS platforms include WordPress, Drupal, and Joomla.

On the other hand, Jekyll is a static site generator that takes plain text files written in markup languages like Markdown or HTML and generates static HTML files that a web server can serve. Jekyll allows you to create websites or blogs with a predefined structure and layout. It does not require a database or server-side processing, as it generates static files that can be hosted on any web server or content delivery network (CDN).

While Jekyll provides powerful features for building static websites and blogs, it does not have the dynamic content management capabilities typically associated with CMS platforms, such as user authentication, content editing interfaces, and database-driven content management.

However, Jekyll can be combined with other tools or services to add dynamic functionality to static websites, such as client-side JavaScript, APIs, or third-party services.

Is Jekyll faster than WordPress?

Jekyll and WordPress are different tools with different use cases, and their performance can vary depending on various factors. Jekyll is generally known for generating static HTML files, which can offer faster page load times than dynamic CMS platforms like WordPress, which dynamically generate HTML pages on the fly.

Jekyll generates static HTML files during the build process, and these files are served as-is to visitors when they request a page. Static content results in faster page load times because no server-side processing is required to generate the HTML on the fly. Additionally, web servers or content delivery networks (CDNs) can easily cache static HTML files, improving performance.

On the other hand, WordPress is a dynamic CMS that uses a server-side scripting language (such as PHP) and a database to generate HTML pages on the fly when a visitor requests a page. CMS introduces additional server-side processing overhead, database queries, and potentially slower page load times, especially for complex websites with large amounts of dynamic content.

However, it's important to note that performance is affected by various factors, including server configuration, caching, content optimization, and more. Both Jekyll and WordPress can be optimized for performance, and the actual performance of a website built with either tool will depend on how it is implemented, hosted, and configured.

In general, if performance is a top priority and you have a relatively simple website with static content, Jekyll may offer faster page load times than WordPress. However, if you require complex dynamic functionality, frequent content updates, or extensive user management, WordPress or another CMS with dynamic capabilities may be more suitable. Ultimately, the choice between Jekyll and WordPress should be based on your specific needs, technical expertise, and performance requirements.

Is Jekyll a Framework?

No, Jekyll is not a framework but a static site generator (SSG). A framework is a pre-prepared library or set of tools that provides a structure or set of conventions for building web applications or websites. Frameworks typically include predefined libraries, modules, or templates that developers can use to build applications or websites more efficiently, often following specific patterns or best practices.

On the other hand, Jekyll is a static site generator that takes plain text files written in markup languages like Markdown or HTML and generates static HTML files that a web server can serve. Jekyll allows you to create websites or blogs with a predefined structure and layout, but it does not provide dynamic functionality or server-side processing like a framework.

Jekyll is primarily used for creating static websites or blogs that do not require server-side processing or databases. It generates static files that can be hosted on any web server or content delivery network (CDN).

While Jekyll provides a structure and a set of conventions for building static websites or blogs, it is not a framework in the traditional sense, as it does not provide dynamic functionality, server-side processing, or libraries for building web applications. However, Jekyll can be combined with other

tools or services, such as JavaScript, APIs, or third-party services, to add dynamic functionality to static websites.

Is Jekyll a Programming Language?

No, Jekyll is not a programming language. Jekyll is a static site generator built using the Ruby programming language. However, Jekyll itself is not a programming language.

Jekyll uses markup languages like Markdown or HTML and a template language called Liquid to generate static HTML files that a web server can serve. Markup languages like Markdown or HTML are used to define web pages content and structure. At the same time, Liquid is a template language that allows you to add dynamic content, conditional logic, and other features to your Jekyll templates.

On the other hand, Ruby is a general-purpose, object-oriented programming language used to build applications and tools, including Jekyll. Jekyll is a Ruby application that leverages the power of Ruby to generate static websites from plain text files written in markup languages.

While Jekyll involves scripting using Liquid for defining templates, it is not a standalone programming language. Jekyll is a tool that uses Ruby and other markup languages to generate static websites but does not have a programming language.

Is Jekyll based on Ruby?

Yes. Jekyll is a static site generator built using the Ruby programming language. Ruby is a general-purpose, object-oriented programming language known for its simplicity and flexibility. Jekyll is a Ruby-based tool that allows users to create static websites or blogs by processing plain text files written in markup languages like Markdown or HTML and generating static HTML files that a web server can serve.

Jekyll uses Ruby to power its templating system, allowing users to define their website's structure, layout, and behaviour using templates. Jekyll also leverages Ruby's powerful features, such as its file handling capabilities, to process input files, apply templates, and generate output files during the build process.

While Jekyll is built on Ruby, it is important to note that users need to be more proficient in Ruby to use Jekyll. Jekyll provides a higher-level abstraction through its Liquid template language, which allows users to define templates using a simple syntax separate from Ruby. However, users with Ruby knowledge can use Jekyll's extensibility features to create custom plugins or modify their behaviour using Ruby code.

Is Jekyll a Ruby Gem?

Yes, Jekyll is a Ruby gem. Ruby gems are packages or libraries written in the Ruby programming language that can be easily installed and used in Ruby applications or projects. Jekyll is a gem that provides a static site generator for creating static websites or blogs.

To use Jekyll, you typically need to install Ruby and RubyGems, the Ruby package manager, on your system. Once you have Ruby and RubyGems installed, you can install Jekyll as a gem using the following command: gem install jekyll.

This command will install Jekyll as a gem on your system, making it available for use in your Ruby applications or projects. After installation, you can use Jekyll to create, configure, and build static websites or blogs from plain text files written in markup languages like Markdown or HTML. Jekyll provides a set of conventions and templates for building static websites, and it can be customized using its configuration files and plugin system.

Is Jekyll using Ruby on Rails?

No, Jekyll is not built on or dependent on Ruby on Rails. Jekyll is a independed tool not directly related to Ruby on Rails.

While both Jekyll and Ruby on Rails are written in Ruby and used for building websites, they serve different purposes and use cases. Jekyll is a static site generator (SSG) that generates static HTML files from plain text files written in markup languages like Markdown or HTML. It is typically used for creating static websites or blogs without dynamic server-side processing or databases. Jekyll generates static files that can be hosted on any web server or content delivery network (CDN).

On the other hand, Ruby on Rails is a full-stack web application framework that follows the model-view-controller (MVC) architectural pattern. Rails provides a set of conventions and tools for building dynamic, database-driven web applications with server-side processing. Rails includes features like an ORM (Object-Relational Mapping) for database interactions, a templating system for rendering views, and a routing system for handling HTTP requests.

While both Jekyll and Ruby on Rails are written in Ruby, they serve different purposes and are used in different contexts. Jekyll is typically used for static websites or blogs, while Ruby on Rails is used for dynamic web applications with server-side processing. They are not directly related to each other and do not have dependencies on each other.

Is Knowledge of Ruby needed to use Jekyll?

While Jekyll is built using Ruby, and some advanced customization or plugin development in Jekyll may require Ruby knowledge, it is unnecessary to have a deep understanding of Ruby to use Jekyll effectively. Jekyll provides a higher-level abstraction through its template language, Liquid, which allows users to define templates using a simple and separate syntax that does not require extensive Ruby knowledge.

Jekyll primarily aims to generate static websites or blogs from plain text files written in markup languages like Markdown or HTML. Users can create, configure, and build static websites with Jekyll by following its conventions and using its built-in features without needing to write Ruby code.

However, having a basic understanding of Ruby can be helpful for customizing Jekyll templates, creating custom plugins, or making advanced configurations. For example, if you want to create a custom layout or modify the behaviour of a Jekyll theme, you may need to use some Ruby code. But even in such cases, you can typically find examples or documentation that provide the necessary Ruby code snippets or configurations without becoming a Ruby expert.

In summary, while knowledge of Ruby can be beneficial for advanced customization or plugin development in Jekyll, it is optional for basic usage of Jekyll to create static websites or blogs. Jekyll provides a higher-level abstraction through its template language, Liquid, which allows users to define templates without extensive Ruby knowledge.

Find more to know:

- Ruby 101
- Jekyll Docs

Is Jekyll using Markdown?

Yes, Jekyll uses Markdown as one of the markup languages to create content for static websites or blogs. Markdown is a lightweight markup language widely used for creating formatted text documents that are easy to read and write. Jekyll supports Markdown as one of its primary markup languages, along with HTML, to create content for web pages.

With Jekyll, users can write content in Markdown files, which are plain text files with a specific syntax that allows easy text formatting and simple markup. Jekyll then processes these Markdown files during the build process and converts them into static HTML files that a web server can serve.

Markdown is a popular choice for content creation in Jekyll because of its simplicity, readability, and compatibility with other tools and platforms. Markdown allows users to write content in a simple, human-readable format converted into HTML by Jekyll during the build process. Markdown makes it easy for users to create formatted content for their Jekyll-powered websites or blogs without needing to write raw HTML code.

Find more to know:

- Jekyll Docs

Is Jekyll using Asciidoc?

Yes, Jekyll supports Asciidoc as one of the markup languages for creating content. Asciidoc is a lightweight markup language designed for creating technical documentation, providing more advanced formatting options than Markdown.

With Jekyll, users can write content in Asciidoc, plain text files with a specific syntax that allows easy text formatting with markup. Jekyll then processes these Asciidoc files during the build process and converts them into static HTML files that a web server can serve.

Asciidoc provides a richer feature set than Markdown, including support for tables, diagrams, footnotes, cross-references, and more. It also allows for more fine-grained control over the output formatting, making it suitable for more complex documentation requirements.

Jekyll supports Asciidoc as an alternative to Markdown, allowing users to choose the markup language that best fits their content creation needs.

Find more to know:

- Asciidoctor

Can I use HTML with Jekyll?

Yes, you can use HTML with Jekyll. Jekyll supports HTML as one of its primary markup languages for creating content. While Jekyll is known for its support of Markdown and Asciidoc, it also allows you to include raw HTML code directly in your Jekyll templates or content files.

Using HTML means writing HTML code directly in your Jekyll templates, layouts, or includes, just like in a regular HTML file. Jekyll will process this HTML code during the build process and generate static HTML files that a web server can serve.

Using HTML directly in your Jekyll templates or content files can be useful in scenarios where you need more fine-grained control over the output HTML or when you want to include custom HTML elements, JavaScript, or other HTML-related functionality in your Jekyll-powered website or blog.

It's worth noting that while Jekyll allows you to use HTML, it also provides higher-level abstraction through its built-in template language, Liquid, which makes it easier to create dynamic content and layouts without needing to write raw HTML code. However, if you are familiar with HTML and prefer to use it directly in your Jekyll templates or content files, you have the flexibility to do so.

Find more to know:

- Jekyll Docs

Can I use CSS with Jekyll?

Yes, you can use CSS with Jekyll. Jekyll is a static site generator that generates static HTML files, which can be styled using CSS like any other HTML web page.

When building a Jekyll-powered website or blog, you can create custom CSS files to define the styles for your site. Using the standard HTML <link> element in your Jekyll templates or layouts, you can include CSS files in your Jekyll project. For example, you can create a CSS file called style.css and include it in your Jekyll template like so:

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <title>My Jekyll Site</title>
  <link rel="stylesheet" href="/assets/css/style.css">
 </head>
 <body>
  <!-- Your content here -->
 </body>
</html>
```

In this example, the style.css file is assumed to be located in the /assets/css/ directory relative to the root of your Jekyll project. You can also use CSS frameworks or libraries in combination with Jekyll to enhance the styling of your site. Include the CSS files of the desired framework or library in your

Jekyll templates or layouts, just like any other CSS file. Jekyll does not have built-in CSS generation or processing capabilities as it does with Markdown or Asciidoc for content. CSS files in Jekyll are static files that are copied as-is to the output directory during the build process, and their content is not processed or transformed by Jekyll. Therefore, you have full control over your CSS and can use any CSS code or techniques compatible with static HTML web pages.

Find more to know:

- Jekyll Docs

Can I use Javascript with Jekyll?

Yes, you can use JavaScript with Jekyll. Jekyll is a static site generator that generates static HTML files. You can include JavaScript in you Jekyll-powered website or blog to add interactivity and dynamic functionality to your site. There are several ways you can use JavaScript with Jekyll

Inline JavaScript

You can include JavaScript directly in your Jekyll templates or content files using the <script> element. For example, you can add JavaScript code directly within a <script> tag in your HTML templates or layouts, just like in any regular HTML web page.

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <title>My Jekyll Site</title>
 </head>
 <body>
  <!-- Your content here -->
 <script>
  // Your JavaScript code here
 </script>
 </body>
</html>
```

External JavaScript

You can include external JavaScript files in your Jekyll project, similar to CSS files. You can link to external JavaScript files using the <script> element with the src attribute pointing to the location of the JavaScript file in your Jekyll project.

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <title>My Jekyll Site</title>
  <script src="/assets/js/script.js"></script>
 </head>
 <body>
  <!-- Your content here -->
 </body>
</html>
```

In this example, the script.js file is assumed to be located in the folder /assets/js/ relative to the root of your Jekyll project. You can also use JavaScript frameworks or libraries in combination with Jekyll to add more complex functionality to your site. Include the JavaScript files of the desired framework or library in your Jekyll templates or layouts, just like you would with any other JavaScript file.

Jekyll does not have built-in JavaScript generation or processing capabilities like it does with Markdown or Asciidoc for content. JavaScript files in Jekyll are static files that are copied as-is to the output directory during the build process, and their content is not processed or transformed by Jekyll.

Therefore, you have full control over your JavaScript code and can use any JavaScript techniques or libraries compatible with static HTML web pages.

Find more to know:

- Jekyll Docs

Can I use Bootstrap with Jekyll?

Yes, you can use Bootstrap with Jekyll. Bootstrap is a popular CSS framework that provides pre-designed UI components and styles for building responsive web pages. Jekyll is a static site generator that generates static HTML files. You can use Bootstrap's CSS and JavaScript files in your Jekyll-powered website or blog to enhance the styling and functionality of your site. Here's how you can use Bootstrap with Jekyll.

Include Bootstrap CSS

You can include Bootstrap's CSS files in your Jekyll project by downloading the CSS files from the Bootstrap website and then adding them to your Jekyll project's CSS directory. Once you've added the Bootstrap CSS files to your Jekyll project, you can include them in your Jekyll templates or layouts using the HTML <link> element.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My Jekyll Site</title>
    <link rel="stylesheet" href="/assets/css/bootstrap.min.css">
  </head>
  <body>
    <!-- Your content here -->
  </body>
</html>
```

Include Bootstrap JavaScript

You can include Bootstrap's JavaScript files in your Jekyll project by downloading the JavaScript files from the Bootstrap website and then adding them to your Jekyll project's JavaScript directory. Once you've added the Bootstrap JavaScript files to your Jekyll project, you can include them in your Jekyll templates or layouts using the HTML <script> element.

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <title>My Jekyll Site</title>
 </head>
 <body>
  <!-- Your content here -->
 <script src="/assets/js/bootstrap.min.js"></script>
 </body>
</html>
```

Use Bootstrap components

With Bootstrap's CSS and JavaScript files included in your Jekyll project, you can now use Bootstrap's pre-designed UI components and styles in your Jekyll templates or content files. Bootstrap includes buttons, forms, navigation bars, modals, and more, which you can add to your Jekyll templates or content files using Bootstrap's class names and markup.

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <title>My Jekyll Site</title>
  <link rel="stylesheet" href="/assets/css/bootstrap.min.css">
 </head>
 <body>
  <!-- Bootstrap component example -->
  <button class="btn btn-primary">Click me</button>
  <!-- Your content here -->
  <script src="/assets/js/bootstrap.min.js"></script>
 </body>
</html>
```

By using Bootstrap with Jekyll, you can easily leverage Bootstrap's responsive design and pre-designed UI components to create modern and visually appealing websites or blogs with Jekyll.

What is Structured Data?

Creating structured data is one of the most powerful and simultaneously least understood technique for Search Engine Optimizing. Structured data for webpages is a programming work, mostly supported by already existing software modules and libraries.

The data required is taken out of the existing source of a webpage. That means no additional content is required but some programming to enhance the already existing content. The data enables search providers to understand how to interpret the content of a webpage by their robots. The additional information enables search engines to have a more human-like understanding of the content to add extended information to the index to support for better-rendered result pages.

Find more on https://jekyll.one/posts/public/articles_seo/seo/2023/07/17/magic-seo-structured-data/

What are Canonical links?

A canonical link or canonical URL allows website operators in an HTML document to identify the original resource with the content used more than once known as duplicate content.

This link type is defined in {RFC 6596} and is assigned as a link element to the canonical attribute, noted in the <head> section of an HTML document.

Example: <link rel="canonical" href="https://example.com/canonical_link/">

Several URIs often refer to the same content. Examples are dynamically generated content, as in forums or others. Content management systems (CMS) or website generators (SSGs) like Jekyll may reference to URLs where the same (or similar) content available is under different addresses.

Linking websites, particularly search engines with their link databases, can thus find the original document for a page and click on it in the search results. The attribute was introduced by search engine operators, above all Google, to make it easier for their web crawlers the primary resource to be identified with certainty.

How to generate canonical links using J1?

For J1 Template it is very easy to specify the URL of the original content by using the canonical tag in the frontmatter of a page or post:

Frontmatter example:

```
---
 title:                Rouge
 tagline:                Code Highlighter
 ...
 canonical:                true
 regenerate:                 false
 permalink:                 /pages/public/learn/roundtrip/rouge/
 ...
---
```

What is J1 Template?

J1 Template is a Bootstrap V5 website template for the static site generator Jekyll. It is a free and open-source project that provides web developers and hobbyists with a clean implementation in Jekyll for starting new projects based on Bootstrap V5.

General Features

- Windows, Linux and MacOS platforms supported
- Jekyll v4 Support
- Ruby v3 Support
- Asciidoc and Markdown Support
- Asciidoctor plugins included
- Bootstrap V5 Support
- Responsive Design
- Responsive Text
- Responsive HTML Tables
- Compressed HTML, CSS and Javascript Support
- Themes from Bootswatch supported (Bootstrap V5)
- Icon Font Support for MDI, FA, and Iconify
- Themeable source code highlighting (Rouge)
- SEO Support
- Auto-Optimization for CSS-, JS, and HTML files
- Desktop and Mobile navigation ready
- Fully configurable, no programming needed
- Near 100/100/100 Google Lighthouse Scores

Modules and Extensions

- Bootstrap extensions included
- Asciidoctor extensions included
- Infinite Scroll Support
- Animate on Scroll Support
- Lazy Load Support for CSS- and Font Icon files
- Support for configurable HTML Selects (msDropdown, slimSelect)
- Full-text Search Engine (Lunr) included
- Master Header Module included
- Cookie Consent Module (GDPR/CPRA compatible OptIn/Out) included
- Clipboard Module included
- Floating Action Menu Buttons (FAMs) included
- Navigation Modules included (Desktop and Mobile)
- Blog Post Navigation included
- Blind Text Support (Lorem ipsum)
- iFrame Support
- Masonry Support
- Lightbox Modules included

- Gallery Modules included
- Carousel Modules included
- Slider Modules included
- Video Player Modules included

Addons and Integrations

- Starter Web including featured example content
- Royalty free Images included
- Comment Provider Support (Disqus and Hyvor)
- Google AdSense Support
- Google Analytics Support (includes OptIn/OptOut support)
- Google Translation Support
- Deploy on Github Pages, Netlify and Heroku

The template combines the best software for the web, and the modules are packed as well. Jekyll One is a full-featured theme for Jekyll to create modern dynamic websites for free. The full source of J1 (as a J1 project) is available at https://github.com/jekyll-one/j1-starter/.

What is the Jamstack?

The Jamstack is a modern web development architecture that decouples the web experience layer from data and business logic, improving flexibility, scalability, performance, and maintainability. It is not a specific technology or framework but a different architecture for building apps and websites.

The term JAM stands for JavaScript, APIs, and Markup (generated by a static site generator). The Jamstack approach allows developers to quickly create and efficiently serve static websites to users. In a Jamstack web application, as much HTML as possible is pre-built and stored in a content delivery network (CDN). This enables a composable architecture for the web where custom logic and 3rd party services are consumed through APIs.

If you're interested in learning more about Jamstack, you can visit the official website at https://jamstack.org/.