Use your own syntax highlighter for code blocks

Last Modified on 07/01/2025 5:15 pm EDT

By default, KnowledgeOwl offers a basic syntax highlighting color scheme:

Sample default code syntax highlighting for HTML

You might want to add a bit more color to your syntax, row numbers, or to match your company branding, maybe something like this:

Sample Prism code syntax highlighting for HTML

To customize the syntax highlighting, add a third-party syntax highlighter.

Below, we walk through using a third-party syntax highlighter called Prism, which we use in this knowledge base. We're not specifically endorsing Prism as a syntax highlighter, just using it as an example!

General setup

Each syntax highlighter may have a slightly different setup process, but generally speaking, there's a two-step setup process:

Step 1: Overall setup

This is a one-time setup to include the syntax highlighter's relevant files in your knowledge base in **Customize > Style (HTML & CSS)**.

Most likely you'll need:

- One or more **scripts** provided as part of the syntax highlighter to complete the highlighting. These are generally in the form of a minified .js file, though they can take other forms.
- One or more **CSS** files (usually provided as part of the syntax highlighter, often customizable) to handle the styling/theme of the highlighter.
- Additional **scripts** may be necessary for specific plugins, languages, or functionality from your syntax highlighter.

You'll also need to disable the default syntax highlighting in **Customize > Website**.

Step 2: Add appropriate class, attribute, or tag in your content

As you add code in your content, add the appropriate formats to trigger the highlighting. Most syntax highlighters use the convention of code needing to be placed in a preformatted block, and then placed within a code block. You can use our WYSIWYG editor to create this overall setup, but you may need to modify the code formatting.

Our built-in code functionality uses data-language attributes, and many syntax highlighters use different attributes. Pay attention to what your syntax highlighter uses and get comfortable hopping into the Code View in KnowledgeOwl to adjust it accordingly.

For example, Prism uses a class that is set to language-xxx where xxx is the language the code is in.

If you used our WYSIWYG, what you'd see for a simple block of HTML code might look like this in Code View:

```
<pre><code data-language="html">
   <!DOCTYPE html>
   <html>
      <head>
         <link rel="stylesheet" href="">
         <script src=""></script>
      </head>
      <body>
         <h1>My First Heading</h1>
         <p>My first paragraph.</p>
      </body>
   </html>
</code></pre>
```

But you'd need to edit it to replace the `data-language="html"` attribute with `class="language-html"`, like this:

```
<pre><code class="language-html">
   <!DOCTYPE html>
   <html>
      <head>
         <link rel="stylesheet" href="">
         <script src=""></script>
      </head>
      <body>
         <h1>My First Heading</h1>
         <p>My first paragraph.</p>
      </body>
   </html>
</code></pre>
```

Example setup: Prism syntax highlighter

To help you understand how to set up a sample syntax highlighter, we'll walk you through how we set up Prism. Note that these steps are specific to Prism and you should follow any instructions provided by your syntax highlighter.

Source file
The steps below include downloading your own Prism JS and CSS files. Prism has many of its files available on CDNJS, so you could also source them from there and just add references to those URLs in your Custom Head.

Step 1: Get your Prism JS and CSS

1. Go to https://prismjs.com.
2. Review the instructions (it's especially helpful to look at the themes and the plugins).
3. Once you have a sense for which options you'd like, select the **Download** option at the top of the page.
4. For our purposes, we're using the minified version, since this keeps it fairly lightweight. You can choose whatever option you'd like.
5. Prism's download gives you the option to select themes, languages, and extra plugins. We use these options:
    a. **Theme:** Tomorrow Night (because dark mode!)
    b. **Languages:** We kept the defaults (Markup + HTML + XML + SVG + MathML + SSML + Atom + R, CSS, C-like, and JavaScript) and also added JSON + Web App Manifest, Bash (which works for cURL highlighting), and PHP, since we occasionally show JSON examples and PHP examples.
    c. **Extra plugins:** Use what seems good to you. We like line numbers, inline color, toolbar, and copy to clipboard button--those are all in-use here in our knowledge base. 😉
6. At the bottom of the page, select **Download JS** to download the JS files.
7. Select **Download CSS** to download your CSS file.
8. If you'd like to customize the color choices in your theme, edit the prism.css file directly. We tweaked ours a bit so that the syntax highlighting uses color hexes from our branded color palette, but that's not necessary! Just be sure you resave your css file if you do make changes.

Step 2: Add your Prism JS and CSS to KnowledgeOwl

Source file
Prism has many of its files available on CDNJS, so you could also source them from there and just add references to those URLs in your Custom Head. That page includes links to copy the script files directly, but you'd need to select all the files you needed--each language, plugin, etc.

1. In KnowledgeOwl, select **Files** from the left menu. The Files page opens.

2. Select **Add files** and upload your prism.js and prism.css files.
3. Once you've added them, you'll need the URL for each file. Refer to Find a file's URL for more information. We recommend copying and pasting the URLs to a text editor for now.
4. Go to **Customize** > **Style (HTML & CSS)**.
5. In the **Customize HTML, CSS, and JS** section, select **Custom <head>**.
6. Copy the code below and paste it into your Custom head:

```
<link rel="stylesheet" href="">
<script src=""></script>
```

7. Replace the stylesheet href URL in row 1 with the URL of your prism.css file, copied in step 4.
8. Replace the src URL in row 2 with the URL of your prism.js file, copied in step 4.
9. **Save** your changes.

Step 3: Disable default syntax highlighting

1. Go to **Customize** > **Website**.
2. In the **Links and behavior** section, under **Content**, check the box to **Disable the default code syntax highlighter**.
3. **Save** your changes.

Step 4/ongoing: Use the correct format in your code blocks

As mentioned above, Prism will only apply syntax highlighting to code that's in a preformatted text section, in a code block, with class set to language-xxx where the language references a code/format it understands.

Here's a sample of the code block format:

```
<pre><code class="language-xxx">
   Actual code goes here
</code></pre>
```

Where `language-xxx` is replaced by a valid language code in Prism's formatting, like `language-html` or `language-css`. Refer to Prism's Supported Languages section for more information on the supported language codes. They do loosely align to the code data-language our WYSIWYG uses.

Preformatted text that doesn't have a code block in it--or code blocks that lack the `class="language-xxx"` format--

will not be highlighted.

The fastest way to get to this setup using our WYSIWYG is to:

1. Paste your code into the editor.
2. Highlight the code and select the Preformatted Text option in the formatting dropdown (or use Ctrl + Alt + 7).
3. With the code still highlighted, use the Code Block control to add a Code block with whatever language formatting you want.
4. Then shift into Code View to edit the code to change data-language="whatever" to the format your syntax highligher uses.

Some plugins, like the line numbers plugin, require additional formatting.

For the easiest use of the line numbers plugin, we found that editing **Customize > Style (HTML & CSS) > Custom HTML > Article** to add the "line-numbers" class to the div that contains the article body template worked easiest, as that automatically adds line-numbers to all blocks of code, like this in row 10:

```
<div class="hg-article-body line numbers">
   [article("body")]
   [article("required_reading_acknowledgement")]
</div>
```