

## Theme Documentation - Content

9Dillon included in documentation

05.03.2020 1726 words 9 minutes



Find out how to create and organize your content quickly and intuitively in **Lovelit** theme.

### # 1 Contents Organization

A few suggestions to help you get a good looking site quickly:

- Keep post pages in the `content/posts` directory, for example: `content/posts/my-first-post.md`
- Keep other pages in the `content` directory, for example: `content/about.md`
- Local resources organization

#### Local Resource Reference

NEW 0.2.10

There are three ways to reference local resources such as **images** and **music**:

1. Using **page resources** in **page bundles**. You can reference page resources by the value for `Resources.GetMatch` or the filepath of the resource relative to the page directory directly.
2. Store resources in the **assets** directory, which is `/assets` by default. The filepath of the resource to reference in the post is relative to the assets directory.
3. Store resources in the **static** directory, which is `/static` by default. The filepath of the resource to reference in the post is relative to the static directory.

The **priority** of references is also in the above order.

There are many places in the theme where the above local resource references can be used, such as **links**, **images**, `image` shortcode, `music` shortcode and some params in the **front matter**.

Images in page resources or assets directory processing will be supported in the future. It's really cool! 😊

### # 2 Front Matter

**Hugo** allows you to add front matter in `yaml`, `toml` or `json` to your content files.

#### Note

**Not all** of the below front matters need to be set in each of your posts. It is necessary only if the front matters and the `page` part in your **site configuration** are inconsistent.

Here is a front matter example:

#### > YAML

- **title**: the title for the content.
- **subtitle**: NEW 0.2.0 the subtitle for the content.
- **date**: the datetime assigned to this page, which is usually fetched from the `date` field in front matter, but this behaviour is configurabl in the **site configuration**.
- **lastmod**: the datetime at which the content was last modified.
- **draft**: if `true`, the content will not be rendered unless the `--buildDrafts` / `-D` flag is passed to the `hugo` command.
- **author**: the author for the content.
- **authorLink**: the link of the author.
- **description**: the description for the content.
- **license**: the special lisenze for this content.
- **images**: page images for Open Graph and Twitter Cards.
- **tags**: the tags for the content.
- **categories**: the categories for the content.
- **featuredImage**: the featured image for the content.
- **featuredImagePreview**: the featured image for the content preview in the home page.
- **hiddenFromHomePage**: if `true`, the content will not be shown in the home page.
- **hiddenFromSearch**: NEW 0.2.0 if `true`, the content will not be shown in the search results.
- **twemoji**: NEW 0.2.0 if `true`, the content will enable the twemoji.
- **lightgallery**: if `true`, images in the content will be shown as the gallery.
- **ruby**: NEW 0.2.0 if `true`, the content will enable the ruby extended syntax.
- **fraction**: NEW 0.2.0 if `true`, the content will enable the fraction extended syntax.
- **fontawesome**: NEW 0.2.0 if `true`, the content will enable the Font Awesome extended syntax.
- **linkToMarkdown**: if `true`, the footer of the content will be shown the link to the original Markdown file.
- **rssFullText**: NEW 0.2.4 if `true`, the full text content will be shown in RSS.
- **toc**: CHANGED 0.2.9 the same as the `params.page.toc` part in the site configuration.
- **code**: NEW 0.2.0 the same as the `params.page.code` part in the site configuration.
- **math**: CHANGED 0.2.0 the same as the `params.page.math` part in the site configuration.
- **mapbox**: NEW 0.2.0 the same as the `params.page.mapbox` part in the site configuration.
- **share**: the same as the `params.page.share` part in the **site configuration**.
- **comment**: CHANGED 0.2.0 the same as the `params.page.comment` part in the site configuration.
- **library**: NEW 0.2.7 the same as the `params.page.library` part in the site configuration.

#### CONTENTS

- | 1 Contents Organization
- | 2 Front Matter
- | 3 Content Summaries
  - | Automatic Summary Splitting
  - | Manual Summary Splitting
  - | Front Matter Summary
  - | Use Description as Summary
  - | Priority Order of Summary Selection
- | 4 Basic Markdown Syntax
- | 5 Extended Markdown Syntax
  - | Emoji Support
  - | Mathematical Formula
    - | Block Formula
    - | Inline Formula
  - | Copy-text
  - | mhchem
- | Ruby Annotation
- | Fraction
- | Font Awesome
- | Escape character

- **seo:** **NEW 0.2.10** the same as the `params.page.seo` part in the site configuration.



Tip

**NEW 0.2.10**

**featuredImage** and **featuredImagePreview** support the complete usage of local resource references.

If the page resource with `name: featured-image` or `name: featured-image-preview` is set in the front matter, it is not necessary to set the parameter `featuredImage` or `featuredImagePreview`:

▼ **YAML**

```
1 resources:
2 - name: featured-image
3   src: featured-image.jpg
4 - name: featured-image-preview
5   src: featured-image-preview.jpg
```

### # 3 Content Summaries

**LoveIt** theme uses the summary of the content to display abstract information in the home page. Hugo can generate summaries of your content.



#### Theme Documentation - Basics

🕒 **Dillon** published on 2020-03-06 included in 📁 Documentation

Discover what the Hugo - LoveIt theme is all about and the core-concepts behind it.

[Read More](#)

🔗 installation, configuration

Summary Preview

#### | Automatic Summary Splitting

By default, Hugo automatically takes the first 70 words of your content as its summary.

You may customize the summary length by setting `summaryLength` in the [site configuration](#).

Character Language Feature

If you are creating content in a **CJK** language and want to use Hugo's automatic summary splitting, set `hasCJKLanguage` to `true` in your [site configuration](#).

#### | Manual Summary Splitting

Alternatively, you may add the `<!--more-->` summary divider where you want to split the article.

Content that comes before the summary divider will be used as that content's summary.

🔗 **Note**

Be careful to enter `<!--more-->` exactly; i.e., all lowercase and with no whitespace.

#### | Front Matter Summary

You might want your summary to be something other than the text that starts the article. In this case you can provide a separate summary in the `summary` variable of the article front matter.

#### | Use Description as Summary

You might want your description in the `description` variable of the article front matter as the summary.

You may add the `<!--more-->` summary divider at the start of the article. Keep content that comes before the summary divider empty. Then **LoveIt** theme will use your description as the summary.

#### | Priority Order of Summary Selection

Because there are multiple ways in which a summary can be specified it is useful to understand the order. It is as follows:

1. If there is a `<!--more-->` summary divider present in the article but no content is before the divider, the description will be used as the summary.
2. If there is a `<!--more-->` summary divider present in the article the text up to the divider will be provided as per the manual summary split method.
3. If there is a summary variable in the article front matter the value of the variable will be provided as per the front matter summary method.
4. The text at the start of the article will be provided as per the automatic summary split method.

🔗 **Note**

It is not recommended to include rich text block elements in the summary, which will cause typographic errors. Such as code blocks, pictures, tables, etc.

### # 4 Basic Markdown Syntax

This part is shown in the [basic markdown syntax page](#).

### # 5 Extended Markdown Syntax

**LoveIt** theme has some extended syntax elements for you to write articles.

#### | Emoji Support

This part is shown in the [emoji support page](#).

#### | Mathematical Formula

**LoveIt** theme supports mathematical formulas based on  $K^ATpX$ .

Set the property `enable = true` under `[params.math]` in your [site configuration](#) and the property `math: true` of the article front matter to enable the automatic rendering of mathematical formulas.

🔗 **Tip**

Here is a list of  $TpX$  functions supported by  $K^ATpX$ .

#### | Block Formula

The default block delimiters are `$$ / $$` and `\\[ / \\]` :

▼ Markdown

```
1 $$ c = \pm\sqrt{a^2 + b^2} $$
2
3 \\[ f(x)=\int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x} d\xi ]]
```

The rendered output looks like this:

$$c = \pm \sqrt{a^2 + b^2}$$
$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x} d\xi$$

### Inline Formula

The default inline delimiters are `$ / $` and `\\( / \\)` :

▼ Markdown

```
1 $ c = \pm\sqrt{a^2 + b^2} $ and \\( f(x)=\int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x} d\xi )
```

The rendered output looks like this:

$$c = \pm \sqrt{a^2 + b^2} \text{ and } f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x} d\xi$$

💡 Tip

You can add more block and inline delimiters in your [site configuration](#).

### Copy-tex

**Copy-tex** is an extension for **KaTeX**.

By the extension, when selecting and copying **KaTeX** rendered elements, copies their **LaTeX** source to the clipboard.

Set the property `copyTex = true` under `[params.math]` in your [site configuration](#) to enable Copy-tex.

Select and copy the formula rendered in the previous section, and you can find that the copied content is the LaTeX source code.

### mhchem

**mhchem** is an extension for **KaTeX**.

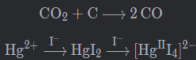
By the extension, you can write beautiful chemical equations easily in the article.

Set the property `mhchem = true` under `[params.math]` in your [site configuration](#) to enable mhchem.

▼ Markdown

```
1 $$ \ce{CO2 + C -> 2 CO} $$
2
3 $$ \ce{Hg^{2+} ->[I-] HgI2 ->[I-] [Hg^{II}I4]^{2-}} $$
```

The rendered output looks like this:



### Ruby Annotation

An extended Markdown syntax for **ruby annotation** is supported in **LoveIt** theme:

▼ Markdown

```
1 [Hugo]^(An open-source static site generator)
```

The rendered output looks like this:

An open-source static site generator  
**Hugo**

### Fraction

**NEW 0.2.0**

An extended Markdown syntax for **fraction** is supported in **LoveIt** theme:

▼ Markdown

```
1 [Light]/[Dark]
2
3 [99]/[100]
```

The rendered output looks like this:

Light/Dark  
90/100

### Font Awesome

**LoveIt** theme uses **Font Awesome** as the icon library. You can easily use these icons in your articles.

Get the `class` of icons you wanted from the [Font Awesome website](#).

▼ Markdown

```
1 Gone camping! :(fas fa-campground fa-fw): Be back soon.
2
3 That is so funny! :(far fa-grin-tears):
```

The rendered output looks like this:

Gone camping! 🏕️ Be back soon.

That is so funny! 😄

### Escape character

In some special cases (when writing this theme documentation 📖), your content will conflict with basic or extended Markdown syntax, and it is inevitable.

The escape character syntax can help you build the content you wanted:

▼ Markdown

```
1 {?X} -> X
```

For example, two `:` will enable emoji syntax, which is not the behavior you want. The escape character syntax is like this:

▼ Markdown

```
1 {?:}joy:
```

The rendered output looks like this:

:joy: instead of 😄

💡 Tip

This is related to [an issue for Hugo](#), which has not been resolved.

Another example is:

▼ Markdown

`<img alt="A placeholder image with a broken image icon." data-bbox="300 65 435 75"/>`