Configuration

2022-01-02 00:00:00.000000000 -03:00

Contents

Before starting

Enable the plugin

Add the plugin to Jekyll's `_config.yml` file:

```
plugins:
- jekyll-printing-press
```

Configuration

2022-01-02 00:00:00.000000000 -03:00

Contents

Before starting 1
Enable the plugin 1
Configure the plugin 2

Before starting

Enable the plugin

Add the plugin to Jekyll's _config.yml file:

plugins:
- jekyll-printing-press

Configure the plugin

Plugin-specific configuration goes under a `pandoc` key on `_config.yml`:

```yaml
pandoc:
  # Configuration goes here
```

Use Pandoc's markdown flavor for generating Jekyll pages

Replace `kramdown` as default Markdown parser on Jekyll:

```yaml
markdown: Pandoc
```

Markdown content on the Jekyll site is processed as Liquid (unless disabled) and then rendered using the HTML5 output format from Pandoc.

You can configure this options later on.

Formats and options

Pandoc options can be configured on Jekyll's side by adding them as YAML key-values. You can either use snake_case or kebab-case.

For more info on options and possible values, please consult Pandoc's documentation.

Add an `options` key on the configuration file.

Configure the plugin

Plugin-specific configuration goes under a `pandoc` key on `_config.yml`:

```yaml
pandoc:
  # Configuration goes here
```

Use Pandoc's markdown flavor for generating Jekyll pages

Replace `kramdown` as default Markdown parser on Jekyll:

```yaml
markdown: Pandoc
```

Markdown content on the Jekyll site is processed as Liquid (unless disabled) and then rendered using the HTML5 output format from Pandoc.

You can configure this options later on.

Formats and options

Pandoc options can be configured on Jekyll's side by adding them as YAML key-values. You can either use snake_case or kebab-case.

For more info on options and possible values, please consult Pandoc's documentation.

Add an `options` key on the configuration file.

```yaml
pandoc:
  options:
```

Then, add a key for any output format.

```yaml
pandoc:
  options:
    html5:
    pdf:
    epub:
```

Each key will contain an option and their values for a specific format.

```yaml
pandoc:
  options:
    html5:
      table-of-contents: true
      toc-depth: 3
    pdf:
      variable:
      - documentclass=book
      - papersize=a5
    epub: true
```

If you want to enable a format without passing any option, just add `true` as its value, like in the `epub` example.

Some options don't need a value, so they can also be enabled with a `true` value. Others do require a value, like in `toc-depth` for `html5`, or accept a list of values, like in `variable` for `pdf`.

```yaml
pandoc:
  options:
```

Then, add a key for any output format.

```yaml
pandoc:
  options:
    html5:
    pdf:
    epub:
```

Each key will contain an option and their values for a specific format.

```yaml
pandoc:
  options:
    html5:
      table-of-contents: true
      toc-depth: 3
    pdf:
      variable:
      - documentclass=book
      - papersize=a5
    epub: true
```

If you want to enable a format without passing any option, just add `true` as its value, like in the `epub` example.

Some options don't need a value, so they can also be enabled with a `true` value. Others do require a value, like in `toc-depth` for `html5`, or accept a list of values, like in `variable` for `pdf`.

These options and values can be as long as you need them to be. Any option supported by Pandoc can go here.

Options for site publication When you enable Pandoc for generating HTML pages, you can set its options using the `html5` format.

```
pandoc:
  options:
    html5:
      # Web pages options go here
```

Common options Sometimes you need to share options between formats. In this case you can use a special `common` key which will share options between all formats.

In this example, all three formats now have a table of contents, but only HTML5 produces one including third-level headers.

```
pandoc:
  options:
    common:
      table-of-contents: true
    html5:
      toc-depth: 3
    pdf:
      variable:
      - documentclass=book
      - papersize=a5
    epub: true
```

These options and values can be as long as you need them to be. Any option supported by Pandoc can go here.

Options for site publication When you enable Pandoc for generating HTML pages, you can set its options using the `html5` format.

```
pandoc:
  options:
    html5:
      # Web pages options go here
```

Common options Sometimes you need to share options between formats. In this case you can use a special `common` key which will share options between all formats.

In this example, all three formats now have a table of contents, but only HTML5 produces one including third-level headers.

```
pandoc:
  options:
    common:
      table-of-contents: true
    html5:
      toc-depth: 3
    pdf:
      variable:
      - documentclass=book
      - papersize=a5
    epub: true
```

Disabling or changing common options If you need to disable a common option on a format, but keep it for others, set its value to `false`.

```
pandoc:
  options:
    common:
      table-of-contents: true
    html5:
      toc-depth: 3
    pdf:
      variable:
      - documentclass=book
      - papersize=a5
    epub:
      table-of-contents: false
```

Now only `html5` and `pdf` have TOCs, and `epub` doesn't.

Same if you want to set a different value:

```
pandoc:
  options:
    common:
      table-of-contents: true
      toc-depth: 3
    html5:
      toc-depth: 4
    pdf:
      variable:
      - documentclass=book
```

Disabling or changing common options If you need to disable a common option on a format, but keep it for others, set its value to `false`.

```
pandoc:
  options:
    common:
      table-of-contents: true
    html5:
      toc-depth: 3
    pdf:
      variable:
      - documentclass=book
      - papersize=a5
    epub:
      table-of-contents: false
```

Now only `html5` and `pdf` have TOCs, and `epub` doesn't.

Same if you want to set a different value:

```
pandoc:
  options:
    common:
      table-of-contents: true
      toc-depth: 3
    html5:
      toc-depth: 4
    pdf:
      variable:
      - documentclass=book
      - papersize=a5
    epub: true
```

Now all formats have a table of contents with three-level headers, except `html5` where the TOC is four-level deep.

```
      - papersize=a5
    epub: true
```

Now all formats have a table of contents with three-level headers, except `html5` where the TOC is four-level deep.