Main Title

Subtitle

Author1, Author2

Revision Date

Table of Contents

1. Rake Jekyll Asciidoctor PDF

This gem allows you to generate PDFs from your Asciidoctor files.

[Gem Version]

To simplify installation, Asciidoctor is packaged and distributed as a RubyGem (aka gem) to RubyGems.org.

1.1. Bundler

1. Open your Gemfile in the root folder of your Jekyll project
2. Add the `rake_jekyll_asciidoctor_pdf` gem to your Gemfile as follows:

   ```
   source 'https://rubygems.org'
   gem 'rake_jekyll_asciidoctor_pdf'

   # or specify the version explicitly
   # gem 'rake_jekyll_asciidoctor_pdf', '0.0.11'
   ```

3. Save the Gemfile
4. Open a terminal and install the gem using:

   ```
   $ bundle
   ```

1.2. Rakefile

1. Open your Rakefile in the root folder of your Jekyll project
2. Add following lines

```
require 'rake_jekyll_asciidoctor_pdf'

RakeJekyllAsciidoctorPdf::BuildTask.new('buildAll') do |t|
  t.jekyll_config_file = '_config.yml'
end
```

1. Save the Rakefile

1.3. Jekyll Configuration

1. Open your _config.yml in the root folder of your Jekyll project
2. Add following lines

```
jap_config:
    sidebar_file: '_data/sidebar.yml'
    output_path: 'pdfs'
    source_path: '.'
    assets_list: ['media']
    exclude_list: []
    fullsite:
        title: "Main Title"
        subtitle: "Subtitle"
        authors: "Author1, Author2"
        revision: "Revision Date"
        background_image: 'cover_page_background.jpg'
    pdf_theme:
        extends: default
        role:
          love:
            font-color: E0245E
          wild:
            font-color: 222222
            text-transform: uppercase
            font-style: italic
            background-color: BCD637
            border-offset: 1.5
        header:
          font_color: 999999
          height: 0.5in
          recto:
            columns: "<40% =20% >40%"
            right:
              content: '_{organization}_'
            center:
              content: image:media/sample-logo.jpg[pdfwidth=0.25in]
          verso:
            columns: $header_recto_columns
            left:
              content: $header_recto_right_content
            center:
              content: $header_recto_center_content
        footer:
          height: 0.45in
          recto:
            right:
              content: '{section-or-chapter-title} | {page-number}'
          verso:
            left:
              content: '{page-number} | {chapter-title}'
```

This configuration assume an specific directory structure

1.4. Directory Structure

A basic Jekyll site usually looks something like this:

```
.
├───── _config.yml
├───── _data
│     └───── sidebar.yml
├───── media
│     └───── assets files
├───── pdfs # this folder will be created by rake_jekyll_asciidoctor_pdf
│     ├───── pages/
│     └───── fullsite/
└───── *.adoc # all asciidoc files are in the root level
```

1.5. Usage

```
$ bundle exec rake buildAll
```

1.6. Links

- AsciiDoctor-PDF
- rake_jekyll_asciidoctor_pdf library
- Rake-Jekyll

2. Documentation Theme for Jekyll

This is a demonstration of {library}. And this is the preamble of this document. ✓

Sign Name code Decimal code Hex code
☑ ☑ ☑
□ □ □
□ □ □

Purpose

This document exercises many of the features of AsciiDoc to test the {library} implementation.

💡 If you want the output to look familiar, copy (or link) the AsciiDoc stylesheet, asciidoc.css, to the output directory.

ℹ Items marked with TODO are either not yet supported or a work in progress.

2.1. First Steps with AsciiDoc

Inline markup

- single quotes around a phrase place 'emphasis'
- astericks around a phrase make the text **bold**
- double astericks around one or more **l**etters in a word make those letters bold
- double underscore around a *sub*string in a word emphasize that substring
- use carrots around characters to make them $^{\text{super}}$script
- use tildes around characters to make them $_{\text{sub}}$script
- to pass through <u>HTML</u> directly, surround the text with triple plus
- characters can be escaped using a \
  - for instance, you can escape a quote inside emphasized text like 'Here's Johnny!'
- you can safely use reserved XML characters like <, > and &, which are escaped when rendering
- force a space between inline elements using the {sp} attribute
- hold text together with an intrinsic non-breaking space attribute, {nbsp}
- handle words with unicode characters like in the name Gregory Romé
- claim your copyright ©, registered trademark ® or trademark ™

You can write text with inline links, optionally using an explicit link prefix. In either case, the link can have a query string.

If you want to break a line
just end it in a ▯ sign
and continue typing on the next line.

2.1.1. Lists Upon Lists

Adjacent lists

- this list
- should join
- to have
- four items

Numbered lists

1. These items
2. will be auto-numbered
   a. and can be nested
3. A numbered list can nest
   - unordered
   - list
   - items

Statement

I swear I left it in 'Guy's' car. Let's go look for it.

term

 definition
line two

another term

another definition, which can be literal (indented) or regular paragraph

This should be a standalone paragraph, not grabbed by the definition list.

- first level
  written on two lines
- first level

  ```
  with this literal text
  ```

  - second level
    - third level
      - fourth level

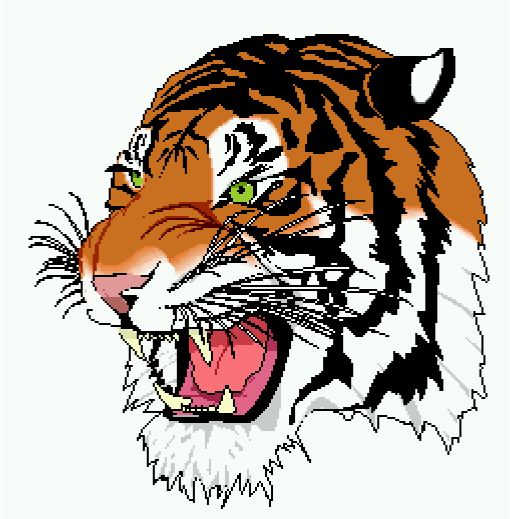
- back to
  first level

Let’s make a horizontal rule…

---

then take a break.

2.2. We’re back!

Want to see a  ?

Do you feel safer with the tiger in a box?

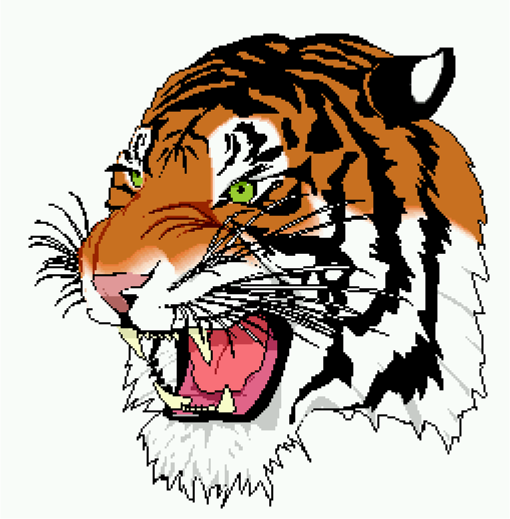


Figure 1. Tiger in a box

Listing 1. Asciidoctor usage example, should contain 3 lines

```
doc = Asciidoctor::Document.new("*This* is it!", :header_footer => false)

puts doc.render
```

Here’s what it outputs (using the built-in templates):

```
<div class="paragraph">
  <p><strong>This</strong> is it!</p>
</div>
```

2.2.1. ``Quotes''

> AsciiDoc is 'so' **powerful**!

This verse comes to mind.

> La la la

Here’s another quote:

> When you have eliminated all which is impossible, then whatever remains, however improbable, must be the truth.
>
> — Sir Arthur Conan Doyle, The Adventures of Sherlock Holmes

2.3. Getting Literal

```
Want to get literal? Just prefix a line with a space (just one will do).
```

```
I'll join that party, too.
```

We forgot to mention in Numbered lists that you can change the numbering style.

a. first item (yeah!)

b. second item, looking `so mono`

c. third item, mono it is!

2.4. Wrap-up

AsciiDoc is quite cool, you should try it!

Info

Go to this URL to learn more about it:

- http://asciidoc.org

Or you could return to the First Steps or Purpose.

Here’s a reference to the definition of another term, in case you forgot it.

> ℹ One more thing. Happy documenting!

When all else fails, head over to http://google.com.

3. Installation

3.1. Directory Structure

You can follow our example repository in the next link

GitHub Repository

4. Configuration

4.1. Directory Structure

You can follow our example repository in the next link

GitHub Repository