

Asciidoctor

Dan Allen, Sarah White, Ryan Waldron

Table of Contents

- The Big Picture 1
- AsciiDoc Processing..... 2
- Requirements 2
- Installation 3
 - (a) gem install 3
 - (b) Bundler..... 4
 - (c) Linux package managers 4
 - Other installation options 6
- Usage 6
 - Command line interface (CLI) 6
 - Ruby API 7
- Contributing 8
- Getting Help 9
- Copyright and Licensing..... 9
- Authors..... 9

[Asciidoctor](#)は 高速な テキストプロセッサで [AsciiDoc](#) をHTML5, DocBook 5(4.5)や他のフォーマットに変換するツールチェーンを配布しています. AsciidoctorはRubyで書かれており, RubyGemとしてパッケージされ, [RubyGems.org](#) で配布されています. gemはいくつかのLinuxディストリビューション, Fedora, Debian, Ubuntuにも含まれています. Asciidoctorはオープンソース [hosted on Github](#) で [the MIT licence](#)のもとに配布されます.

Translations of the document are available in the following languages:

- [汉语](#)
- [English](#)
- [Français](#)

Key documentation

- [What is AsciiDoc?](#)
- [AsciiDoc Writer's Guide](#)
- [AsciiDoc Syntax Reference](#)
- [Asciidoctor User Manual](#)


Rubyの行く先, Asciidoctorの追うところ

AsciidoctorはJRubyを用いてJVM上でも実行できます. Javaや他のJVM言語からAsciidoctor APIを直接呼び出すには, [AsciidoctorJ](#) を使ってください. AsciidoctorJに基づいた, AsciidoctorプロセッサをApache Maven, GradleやJavadocに統合するプラグインがあります.

AsciidoctorはJavaScriptでも実行可能です. [Asciidoctor.js](#), WebブラウザやNode.jsのようなJavaScript環境で動くAsciidoctorの完全機能版, を生成するために, RubyのソースをJavaScriptにトランスパイルするのに [Opal](#)を使います. Asciidoctor.jsはChrome, Atom, Brackets や他のウェブベースのツールの拡張機能としてAsciiDocのプレビューのために使われます.

The Big Picture

Asciidoctorは下図の左側のパネルに示されるように, 平文で書かれた内容を読み, 右のパネルに描かれるようにHTML5に変換します. Asciidoctorは枠にとらわれない快適なエクスペリエンスのためにデフォルトスタイルシートをHTML5時メントに適用します.

<pre>= AsciiDoc is Writing Zen Doc Writer <doc.writer@example.com> :icons: font _Zen_ in the *art* of writing `plain text` with http://asciidoc.org[AsciiDoc]. [TIP] Use http://asciidoctor.org[Asciidoctor] for the best AsciiDoc experience.footnote:[Not to mention the best looking output!] Then icon:twitter[role=aqua] about it! == Sample Section [square] * item 1 * item 2 [source,ruby] ---- puts "Hello, World!" ----</pre>	<div> <h1>AsciiDoc is Writing Zen</h1> <p>Doc Writer – doc.writer@example.com</p> </div> <div> <p>Zen in the art of writing plain text with AsciiDoc.</p> <div>  <p>Use Asciidoctor for the best AsciiDoc experience.^[1] Then tweet about it!</p> </div> <h2>Sample Section</h2> <ul style="list-style-type: none"> ■ item 1 ■ item 2 <pre>puts "Hello, World!"</pre> <hr/> <p>¹ Not to mention the best looking output!</p> </div>
AsciiDoc source	Rendered HTML

AsciiDoc Processing

AsciidoctorはAsciiDoc文法で書かれたテキストを読み込み解釈し、それからHTML5、DocBook5(4.5)やman(ual)を出力するために内蔵コンバータセットにパースツリーを渡します。生成された出力をカスタマイズ、あるいは追加のフォーマットをつくるためにあなた自身のコンバータを使うことや [Tilt-supported](#) テンプレートを読み込むオプションがあります。



AsciidoctorはオリジナルのAsciiDoc Pythonプロセッサ([asciidoc.py](#))の完全互換です。Asciidoctorテストスイートは [> 1,600 tests](#) をAsciiDoc文法との互換性を保証するために有しています。

クラシックなAsciiDoc文法に加えて、Asciidoctorは追加のマークアップとフォントベースのicons(例えば、[icon:fire\[\]](#))などのフォーマッティングオプションとUIエレメント([button:\[Save\]](#))を受け付けます。AsciidoctorはHTML5出力をスタイルするため、モダンで、[Foundation](#) に基づいたレスポンシブテーマをも提供します。

Requirements

AsciidoctorはLinux、OS X (Mac)とWindowsで動き、下記の [Ruby](#)実装の一つを必要とします。

- MRI (Ruby 1.8.7, 1.9.3, 2.0, 2.1, 2.2 & 2.3)
- JRuby (1.7 in Ruby 1.8 and 1.9 modes, 9000)
- Rubinius 2.2.x
- Opal (JavaScript)

もし非英語環境のWindowsを使っているなら, Asciidoctorを起動した時に`Encoding::UndefinedConversionError`に遭遇するでしょう. これを解決するには使っているコンソールの有効なコードページをUTF-8:



```
chcp 65001
```

に変更することを推奨します. 一度この変更をすると, Unicode関連の頭痛の種は消えるでしょう. もしEclipseのようなIDEを使っているなら, 同様にエンコーディングをUTF-8にするのを忘れないでください. AsciidoctorはUTF-8が使われているところで最高の働きを見せます.

Installation

Asciidoctorは (a) `gem install` コマンド, (b) Bundler あるいは (c) 有名Linuxディストリビューションのパッケージマネージャ を用いてインストールされます.



Linuxパッケージマネージャを用いてインストールすることの利点は, もしRubyやRubyGemsライブラリがまだインストールされていなかったら, それら进行处理してくれることです. 欠点はgemのリリース直後にはすぐには有効にならないことです. もし最新バージョンを使いたければ, 必ず `gem` コマンドを使いましょう.

(a) gem install

ターミナルを開, 入力しましょう (先頭の`\$`は除く):

```
$ gem install asciidoctor
```

もし, 先行リリースバージョン(例えばリリース候補版)をインストールしたければ

```
$ gem install asciidoctor --pre
```

アップグレード

もしAsciidoctorの以前のバージョンがインストール済みであれば, 以下によってアップデートできます:



```
$ gem update asciidoctor
```

もし gem update の代わりに **gem install** を使ってgemを新バージョンにした場合, 複数バージョンがインストールされるでしょう. そのときは, 以下のgemコマンドで古いバージョンを削除しましょう:

```
$ gem cleanup asciidoctor
```

(b) Bundler

1. プロジェクトフォルダーのルート(カレントディレクトリ)にGemfileを作成
2. **asciidoctor** gemをGemfileに以下のように追加:

```
source 'https://rubygems.org'
gem 'asciidoctor'
# or specify the version explicitly
# gem 'asciidoctor', '1.5.4'
```

3. Gemfileを保存
4. ターミナルを開き, gemをインストール:

```
$ bundle
```

gemをアップグレードするには, Gemfileで新バージョンを指定し, **bundle** を再び実行してください. **bundle update** は他のgemもアップデートするため推奨されて **いない** ので, 思わぬ結果になるかも知れません.

(c) Linux package managers

DNF (Fedora 21 or greater)

dnfを使いFedora21かそれ以上にインストールするには, ターミナルを開き, 以下を入力してください:

```
$ sudo dnf install -y asciidoctor
```

gemをアップグレードするには:

```
$ sudo dnf update -y asciidoctor
```



お使いのシステムは自動的にrpmパッケージをアップデートするよう設定されているかも知れません.その場合,gemのアップデートのためにあなたがすべきことはありません.

apt-get (Debian, Ubuntu, Mint)

Debian, UbuntuまたはMintにインストールするには,ターミナルを開き,以下を入力してください:

```
$ sudo apt-get install -y asciidoctor
```

gemをアップグレードするには:

```
$ sudo apt-get upgrade -y asciidoctor
```



お使いのシステムは自動的にdebパッケージをアップデートするよう設定されているかも知れません.その場合,gemのアップデートのためにあなたがすべきことはありません.

パッケージマネージャ(apt-get)によってインストールされたバージョンのAsciidoctorは最新リリースのAsciidoctorではないかもしれません.ディストリビューションのリリース毎に,どのバージョンがパッケージされているかはパッケージリポジトリを調べてください.

- [asciidoctor package by Debian release](#)
- [asciidoctor package by Ubuntu release](#)
- [asciidoctor package by Mint release](#)



パッケージマネージャによって管理されているgemをアップデートするのに **gem update** コマンドを使うなどといわれるでしょう.そのようなことをするのは,パッケージマネージャがファイル(/usr/local下にインストールされた)を追跡できなくなるためにシステムが不安定な状態にするためです.単純に,システムgemはパッケージマネージャによってのみ管理されるべきです.

もし,パッケージマネージャによってインストールされたのより新しいバージョンのAsciidoctorを使いたければ,**RVM** や **rbenv** を使ってRubyをホームディレクトリ(すなわち,ユーザースペース)にインストールするべきです.それから,安心して **gem** コマンドをAsciidoctorのアップデート,インストールのために使うことができます.RVMやrbenvを使っているなら,gemはシステムからは孤立した場所にインストールされます.

apk (Alpine Linux)

Alpine Linuxにgemをインストールするには,ターミナルを開き,以下を入力してください:

```
$ sudo apk add asciidoctor
```

gemをアップグレードするには:

```
$ sudo apk add -u asciidoctor
```



お使いのシステムは自動的にapkパッケージをアップデートするよう設定されているかも知れません.その場合, gemのアップデートのためにあなたがすべきことはありません.

Other installation options

- [Installing Asciidoctor using Docker](#)
- [Installing Asciidoctor on Mac OS X](#)

Usage

Asciidoctorのインストールに成功すれば, **asciidoctor** コマンドラインインターフェース(CLI)がPATH中で有効になります. 確認のために, 以下をターミナルで実行しましょう:

```
$ asciidoctor --version
```

AsciidoctorのバージョンとRuby環境についての情報がターミナルに出力されたのを見ることができるはずです.

```
Asciidoctor 1.5.4 [http://asciidoctor.org]  
Runtime Environment (ruby 2.2.2p95 [x86_64-linux]) (lc:UTF-8 fs:UTF-8 in:-  
ex:UTF-8)
```

AsciidoctorはAPIを提供します. APIは他のRubyソフトウェア, Rails, SinatraとGitHub, そして他の言語, Java (via [AsciidoctorJ](#))とJavaScript (via [Asciidoctor.js](#))との統合を意図しています.

Command line interface (CLI)

asciidoctorjs コマンドはAsciidoctorをコマンドライン(つまりターミナル)から起動することを可能にします.

次のコマンドはファイルREADME.adocをHTMLに変換し, 結果を同じディレクトリのREADME.htmlに保存します. 生成されたHTMLファイルの名前はソースファイル依存し, その拡張子を **.html** に変えます.


```
$ asciidoctor README.adoc
```

Asciidoctorプロセッサに様々なフラグやスイッチを与えることで制御できます。それは以下を用いて調べることができます:

```
$ asciidoctor --help
```

例えば、ファイルを異なるディレクトリに書き出すには:

```
$ asciidoctor -D output README.adoc
```

`asciidoctor man page` はコマンドライン・インタフェースの完全なリファレンスを提供します。

`asciidoctor` コマンドの使い方についてもっと学ぶには以下を参照してください。

- [How do I convert a document?](#)
- [How do I use the Asciidoctor stylesheet factory to produce custom themes?](#)

Ruby API

Asciidoctorをアプリケーションの中で使うには、まずgemをrequireする必要があります:

```
require 'asciidoctor'
```

それから、AsciiDocソースファイルをHTMLファイルに変換できます:

```
Asciidoctor.convert_file 'README.adoc', to_file: true, safe: :safe
```



AsciidoctorをAPI経由で使っている時、デフォルトのセーフモードは `:secure` です。セキュアモードでは、`include` ディレクティブを含むいくつかのコア機能は無効化されています。もしこれらの機能を有効化したい場合、明示的にセーフモードを `:server` (推奨)か `:safe` にする必要があります。

AsciiDoc文字列を埋め込みHTML(HTMLページへの挿入)へ変換することもできます:

```
content = '_Zen_ in the art of writing http://asciidoctor.org[AsciiDoc].'  
Asciidoctor.convert content, safe: :safe
```

もし完全なHTMLドキュメントを求めるのであれば、`header_footer` オプションを以下の通り有効にしてください:

```
content = '_Zen_ in the art of writing http://asciidoctor.org[AsciiDoc].'  
html = AsciiDoctor.convert content, header_footer: true, safe: :safe
```

パースされたドキュメントにアクセスしたいのなら、変換を個々のステップに分割することが出来ます:

```
content = '_Zen_ in the art of writing http://asciidoctor.org[AsciiDoc].'  
document = AsciiDoctor.load content, header_footer: true, safe: :safe  
puts document.doctitle  
html = document.convert
```

AsciiDoctorの生成する出力が気に入らないのであれば、あなたはそれを変更できることを忘れないでください! AsciiDoctorはパースされたドキュメントを生成された出力に変換する処理を扱うカスタムコンバーターをサポートしています。

断片的な出力をカスタマイズする簡単な方法の一つはテンプレートコンバーターを使うことです。テンプレートコンバーターによって、ドキュメント中のあらゆるノードの変換を扱うために [Tilt-supported](#) テンプレートファイルを使うことができます。

そのようにすれば、出力を100%制御することができます。APIの使い方や出力のカスタマイズ方法についてのより詳しい情報は [user manual](#) を参照してください。

Contributing

[free software](#) の精神においては、*everyone* がこのプロジェクトを改良するのをたすけることが勧められている。もしエラーや手拔かりをソースコード、ドキュメント、あるいはウェブサイトに見つけたのなら、恥じることなく修正と共に pull request の開設や issue の送信をしてください。New contributors are always welcome!

あなた にもできることがあります:

- 先行バージョン(alpha, beta or preview)の使用
- バグレポート
- 新機能提案
- ドキュメントの執筆
- 仕様の執筆
- コーディング☒—☒パッチでも、足りなすぎるなんてことはありません
 - typoの修正
 - コメントの追加
 - 一貫性のないホワイトスペースの除去
 - テストの記述!

- リファクタリング
- [issues](#) の修正
- パッチの批評

[Contributing](#) ガイドはどうやってスタイルをつくるか, [issue](#)を送るか, 機能リクエスト, コーディング, ドキュメンテーションをAsciidoctor Projectにするかの情報を提供しています.

Getting Help

Asciidoctorプロジェクトはあなたが簡単に著作を書いて, 配布するのをたすけるため開発されています. しかしあなたのフィードバックなしにはできません! ディスカッションリストで, Twitterで, チャットルームで, 質問し, プロジェクトのあらゆる側面について話し合うようお勧めします.

Discussion list (Nabble)

<http://discuss.asciidoctor.org>

Twitter

#asciidoctor hashtag or @asciidoctor mention

Chat (Gitter)

 gitter join chat

GitHub上のAsciidoctorはプロジェクトのソースコード, イシュートラッカー, サブプロジェクトを管理しています.

Source repository (git)

<https://github.com/asciidoctor/asciidoctor>

Issue tracker

<https://github.com/asciidoctor/asciidoctor/issues>

Asciidoctor organization on GitHub

<https://github.com/asciidoctor>

Copyright and Licensing

Copyright © 2012-2016 Dan Allen, Ryan Waldron and the Asciidoctor Project. Free use of this software is granted under the terms of the MIT License.

See the [LICENSE](#) file for details.

Authors

Asciidoctor is led by [Dan Allen](#) and [Sarah White](#) and has received contributions from [many other](#)

individuals in Asciidoctor's awesome community. The project was initiated in 2012 by [Ryan Waldron](#) and based on a [prototype](#) written by [Nick Hengeveld](#).

AsciiDoc was started by Stuart Rackham and has received contributions from many other individuals in the AsciiDoc community.