

# Asciidoctor

Dan Allen, Sarah White, Ryan Waldron

# Table of Contents

- 全局概况 ..... 1
- AsciiDoc Processing..... 2
- 要求 ..... 2
- 安装 ..... 3
  - (a) gem 安装 ..... 3
  - (b) Bundler..... 4
  - (c) Linux 包管理 ..... 4
  - 其他安装选项 ..... 5
- 使用 ..... 5
  - 命令行 (CLI) ..... 6
  - Ruby API ..... 6
- 贡献 ..... 7
- 获得帮助 ..... 8
- 版权和协议 ..... 9
- 作者 ..... 9
- Changelog ..... 9

[Asciidoctor](#) 是一个 快速 文本处理器和发布工具链，它可以将 [AsciiDoc](#) 文档转化成 HTML5、DocBook 5 (或 4.5) 以及其他格式。Asciidoctor 由 Ruby 编写，打包成 RubyGem，然后发布到 [RubyGems.org](#) 上。这个 gem 还被包含道几个 Linux 发行版中，其中包括 Fedora、Debian 和 Ubuntu。Asciidoctor 是开源的，<https://github.com/asciidoctor/asciidoctor>[代码托管在 GitHub]，并且是以 [MIT 协议](#) 授权。

该文档有如下语言的翻译版：

- [English](#)
- [Français](#)

关键文档

- [什么是 AsciiDoctor?](#)
- [AsciiDoc 作家指南](#)
- [AsciiDoc 语法快速参考](#)
- [AsciiDoctor 用户手册](#)


### Ruby 所至，AsciiDoctor 相随

使用 JRuby 可以让 AsciiDoctor 运行在 Java 虚拟机上。使用 [AsciiDoctorJ](#) 就可以让 Java 或者其他 Java 虚拟机语言直接调用 AsciiDoctor API。基于 AsciiDoctorJ 有好多好多插件可用，这些插件可以将 AsciiDoctor 整合到 Apache Maven，Gradle 或 Javadoc 构建中。

AsciiDoctor 也可以运行在 JavaScript 上。我们可以使用 [Opal](#) 将 Ruby 源码编译成 JavaScript 并生成 [AsciiDoctor.js](#)，这是一个全功能版的 AsciiDoctor，可以运行在任意的 JavaScript 环境中，比如 Web 浏览器 或 Node.js。AsciiDoctor.js 被用于 AsciiDoc 预览，支持 Chrome 扩展，Atom，Brackets 或其他基于 Web 的工具。

## 全局概况

AsciiDoctor 以纯文本格式读取内容，见下图左边的面板，将它转换成 HTML5 呈现在右侧面板中。AsciiDoctor 将默认的样式表应用到 HTML5 文档上，提供一个愉快的开箱即用的体验。

<pre> = AsciiDoc is Writing Zen Doc Writer &lt;doc.writer@example.com&gt; :icons: font  _Zen_ in the *art* of writing `plain text` with http://asciidoc.org[AsciiDoc].  [TIP] Use http://asciidoc.org[AsciiDoc] 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 – <a href="mailto:doc.writer@example.com">doc.writer@example.com</a></p> </div> <div> <p>Zen in the <b>art</b> of writing <b>plain text</b> with <a href="http://asciidoc.org">AsciiDoc</a>.</p> <div>  <p>Use <a href="#">AsciiDoc</a> for the best AsciiDoc experience.<sup>[1]</sup> Then <a href="#">❤</a> about it!</p> </div> <h2>Sample Section</h2> <ul style="list-style-type: none"> <li>■ item 1</li> <li>■ item 2</li> </ul> <pre>puts "Hello, World!"</pre> <p><sup>1.</sup> Not to mention the best looking output!</p> </div>
AsciiDoc source	Rendered HTML

# AsciiDoc Processing

AsciiDoctor 读取并处理以 AsciiDoc 语法写作的文件，然后然后将解析出来的解析树交给内置的转化器生成 HTML5，DocBook 5 (或 4.5) 或帮助手册页面输出。 你可以选择使用你自己的转化器或者加载 [Tilt](#) - 支持通过模板来自定义输出或产生附加的格式。



AsciiDoctor 是为了直接替换原 AsciiDoc Python 处理器 ([asciidoc.py](#))。AsciiDoctor 测试套件含有 [> 1,600 测试用例](#) 来确保和 AsciiDoc 语法的兼容性。

除了经典的 AsciiDoc 语法，AsciiDoctor 还添加额外的标记和格式设置选项，例如 font-based 图标（例如：[icon:fire\[\]](#)）和 UI 元素（例如：[button:\[Save\]](#)）。AsciiDoctor 还提供了一个基于 [Foundation](#) 的现代的、响应式主题来美化 HTML5 输出。

## 要求

AsciiDoctor 可以在 Linux，OSX (Mac) 和 Windows，并且需要下面其中一个 [Ruby](#) 实现：

- 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)

我们欢迎你来帮助在这些以及其他平台测试 AsciiDoctor。参考 [Contributing](#) 来学习如何参与进来。

如果你使用一个非英语的 Windows 环境，当调用 Asciidoctor 时，可能会碰到 `Encoding::UndefinedConversionError` 错误。为了解决这个问题，我们建议将控制台的编码更改为 UTF-8：



```
chcp 65001
```

一旦你做了这个改变，所有的编码问题，都将迎刃而解。只要你在任何地方都是 UTF-8，Asciidoctor 总会工作地很好。

## 安装

Asciidoctor 可以通过三种方式安装：（a）使用 `gem install` 命令；（b）使用 Bundler；（c）流行的 Linux 发行版的包管理器



使用 Linux 包管理器安装的好处是如果 Ruby 和 RubyGems 库没有在你的机器上安装，它会一并安装上去。不利的是在 `gem` 发布之后，这类安装包并不是立即可用。如果你需要最新版，你应该总是优先使用 `gem` 命令安装。

### (a) gem 安装

打开一个终端输入如下命令（不含开头的 `$`）：

```
$ gem install asciidoctor
```

如果想安装一个预览版（比如：候选发布版），请使用：

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

#### 升级

如果你安装有一个旧版本的 Asciidoctor，你可以使用下面的命令来升级：

```
$ gem update asciidoctor
```



如果使用 `gem install` 命令来安装一个新版本的 `gem` 来代替升级，则会安装多个版本。如果是这种情况，使用下面的 `gem` 命令来移除旧版本：

```
$ gem cleanup asciidoctor
```

## (b) Bundler

1. 在项目的根目录（或者当前路径），创建一个 **Gemfile** 文件；
2. 在这个文件中添加 **asciidoctor** gem 如下：

```
source 'https://rubygems.org'
gem 'asciidoctor'
# 或者明确指明版本
# gem 'asciidoctor', '1.5.4'
```

3. 保存 **Gemfile** 文件
4. 打开终端，使用如下命令安装 gem：

```
$ bundle
```

要升级 gem 的话，在 **Gemfile** 文件中，指明新版本，然后再次运行 **bundle** 即可。**不推荐** 直接使用 **bundle update** 命令，因为它还会升级其他 gem，也许会造成不可预料的结果。

## (c) Linux 包管理

### DNF (Fedora 21 或更高版本)

在 Fedora 21 或更高版本中安装这个 gem，可以使用 dnf。打开终端并输入如下命令：

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

升级则使用：

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



如果你的 Fedora 系统配置的是自动升级包，在这种情况下，不需要你亲自动手升级。

### apt-get (Debian, Ubuntu, Mint)

在 Debian, Ubuntu 或 Mint 中安装这个 gem，请打开终端并输入如下命令：

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

升级则使用：

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



如果你的 Debian 或 Ubuntu 系统配置的是自动升级包，在这种情况下，不需要你亲自动手升级。

使用包管理器（apt-get）安装的 Asciidoctor 的版本也许不是最新发布版。请查看发行版的包库，来确定每个发行版是打包的哪个版本。

- [Debian 发行版中的 asciidoctor](#)
- [Ubuntu 发行版中的 asciidoctor](#)
- [Mint 发行版中的 asciidoctor](#)



我们建议不要使用 `gem update` 来升级包管理的 gem。这样做会使系统进入不一致的状态，包管理工具将不再跟踪相关文件（通常安装在 `/usr/local` 下。）简单地说，系统的 gem 只能由包管理器进行管理。

如果你想使用一个比包管理器安装的更新版本的 Asciidoctor，你应该使用 [RVM](#) 在你的用户家目录（比如：用户空间）下安装 Ruby。然后，你就可以放心地使用 `gem` 命令来安装或者更新 Asciidoctor gem。当使用 RVM 时，gem 将被安装到与系统隔离的位置。

## apk (Alpine Linux)

在 Alpine Linux 中安装这个 gem，请打开终端并输入如下命令：

```
$ sudo apk add asciidoctor
```

升级则使用：

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



如果你的 Alpine Linux 系统配置的是自动升级包，在这种情况下，不需要你亲自动手升级。

## 其他安装选项

- [使用 Docker 安装 Asciidoctor](#)
- [在 Mac OS X 安装 Asciidoctor](#)

## 使用

如果成功安装 Asciidoctor，则在可执行程序路径中，`asciidoctor` 就可用了。为了验证它的可用性，你可以在终端中执行如下命令：

```
$ asciidoctor --version
```

你应该看到关于 Asciidoctor 和 Ruby 环境信息将打印到你的终端上。

```
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 还提供了一套 API。这套 API 是为了整合其他的 Ruby 软件，例如 Rails、Sinatra、Github，甚至其他语言，比如 Java（通过 [AsciidoctorJ](#)）和 JavaScript（通过 [Asciidoctor.js](#)）。

## 命令行 (CLI)

**asciidoctor** 命令可以让你通过命令行（比如：终端）来调用 Asciidoctor。

下面的命令将 README.adoc 文件转化为 HTML，并且保存到同一目录下的 README.html 文件中。生成的 HTML 文件名源自源文件名，只是将其扩展名改为了 **.html**。

```
$ asciidoctor README.adoc
```

您可以通过添加各种标志和开关控制 Asciidoctor 处理器，通过下面的命令你可以学习它的更多用法：

```
$ asciidoctor --help
```

比如，将文件写入到不同路径里，使用如下命令：

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

**asciidoctor** [帮助页面](#) 提供了这个命令的完整参考。

点击下面的资源，学习更多关于 **asciidoctor** 命令的用法。

- [如何转化文档？](#)
- [如何使用 Asciidoctor 样式工厂来创建自定义主题？](#)

## Ruby API

为了在你应用中使用 Asciidoctor，首先需要引入这个 gem：



```
require 'asciidoctor'
```

然后，你可以通过下面的代码将 AsciiDoc 源文件转化成一个 HTML 文件：

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



当你通过 API 使用 Asciidoctor 时，默认的安全模式是 `:secure`。在 `secure` 模式下，很多核心特性将不可用，包括 `include` 特性。如果你想启用这些特性，你需要明确设置安全模式为 `:server`（推荐）或 `:safe`。

你也可以将 AsciiDoc 字符串转化我内嵌的 HTML（为了插入到一个 HTML 页面），用法如下：

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

如果你想得到完整的 HTML 文档，只需要启用 `header_footer` 选项即可。如下：

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

如果你想访问已经处理过的文档，可以将转化过程拆分成离散的几步：

```
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
```

请注意：如果你不喜欢 Asciidoctor 输出结果，*你完全可以改变它*。Asciidoctor 支持自定义转化器，它可以操作从待处理文件到生成文档整个环节。

一个简单的、细微地自定义输出的方式是使用模板转化器。模板转化器运行你提供一个 [Tilt](#) 模板，这样通过模板文件来操作转化出的文档的每个节点。

这样，你就可以百分之百地控制你的输出。关于更多关于 API 或自定义输出信息，请参考 [用户帮助手册](#)。

## 贡献

自由软件的精神鼓励 *每个人* 来帮助改善这个项目。如果你在源码、文档或网站内容中发现错误或漏洞，请不要犹豫，提交一个议题或者推送一个修复请求。随时欢迎新的贡献者！

这里有几种 **你** 可以做出贡献的方式：

- 使用预发布版本 (alpha, beta 或 preview)
- 报告 Bug
- 提议新功能
- 编写文档
- 编写规范
- 编写任何补丁都不小。
  - 修正错别字
  - 添加评论
  - 清理多余空白
  - 编写测试!
- 重构代码
- 修复 议题
- 审查补丁

[贡献](#) 指南提供了如何提供贡献，包括如何创建、修饰和提交问题、特性、需求、代码和文档给 AsciiDoctor 项目。

## 获得帮助

开发 AsciiDoctor 项目是未来帮助你更容易地书写和发布你的内容。但是，如果没有反馈，我们将寸步难行。我们鼓励你在讨论组、Twitter或聊天室里，提问为题，讨论项目的方方面面，

讨论组 (Nabble)

<http://discuss.asciidoctor.org>

Twitter

#asciidoctor 井号或 @asciidoctor 提醒

聊天 (Gitter)

 gitter join chat

AsciiDoctor 组织在 Github 托管代码、议案跟踪和相关子项目。

代码库 (git)

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

议案跟踪

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

在 *GitHub* 的 *Asciidoctor* 组织

<https://github.com/asciidoctor>

## 版权和协议

Copyright © 2012-2016 Dan Allen, Ryan Waldron and the Asciidoctor Project. 这个软件的免费使用是在MIT 许可条款授予的。

请看 [版权声明](#) 文件来获取更多信息。

## 作者

**Asciidoctor** 由 [Dan Allen](#) 和 [Sarah White](#) 领导，并从 Asciidoctor 社区的 [很多其他独立开发者](#) 上收到了很多贡献。项目最初由 [Ryan Waldron](#) 于 2012 年基于 [Nick Hengeveld](#) 的 [原型](#) 创建。

**AsciiDoc** 由 Stuart Rackham 启动，并从 AsciiDoc 社区的其他独立开发者上收到很多贡献。

## Changelog

请看 [CHANGELOG](#)。