

Asciidoctor

Dan Allen, Sarah White, Ryan Waldron

[AsciiDoctor](#) est un processeur de texte et une chaîne de publication *rapide* et [open source](#) permettant de convertir du contenu [AsciiDoc](#) en HTML5, DocBook, PDF et d'autres formats. AsciiDoctor est écrit en Ruby et fonctionne sur les principaux systèmes d'exploitation. Pour simplifier l'installation, AsciiDoctor est publié au format gem sur [RubyGems.org](#), et il est également disponible en tant que paquet système sur les principales distributions Linux ainsi que sur macOS. AsciiDoctor fonctionne aussi sur la JVM avec [AsciiDoctorJ](#) et dans n'importe quel environnement JavaScript avec [AsciiDoctor.js](#). Le projet AsciiDoctor est [hébergé sur GitHub](#).

Ce document est traduit dans les langues suivantes :

[Anglais](#) | [Chinois](#) | [Japonais](#)

Documentation clé

- [Qu'est ce qu'AsciiDoc ?](#)
- [Guide pour Rédacteur AsciiDoc](#)
- [Syntaxe de Référence AsciiDoc](#)
- [Manuel Utilisateur AsciiDoctor](#)

Sponsors

Nous souhaitons exprimer toute notre reconnaissance à nos généreux sponsors, sans qui AsciiDoctor ne pourrait pas exister. Merci à vous pour votre engagement dans l'amélioration de la documentation technique !

[Okta] [OpenDevise]

Asciidoctor est financé majoritairement par nos **Change Makers**, [Okta](#) et [OpenDevise](#), ainsi que par nos **Strategy Sponsors**, [Khronos Group](#) et Linda Roberts. Un apport financier supplémentaire est assuré par [la communauté](#).

Vous pouvez aider ce projet en devant un sponsor sur [OpenCollective](#).

En un mot

Asciidoctor lit du contenu écrit en texte brut, comme présenté dans la partie gauche de l'image ci-dessous, et le convertit en HTML5, comme présenté dans la partie droite. Asciidoctor applique une feuille de style par défaut au document HTML5 afin de fournir une expérience de lecture agréable, clé en main.

[Prévisualisation d'une source AsciiDoc et le rendu HTML correspondant] |

<https://raw.githubusercontent.com/asciidoctor/asciidoctor/master/screenshot.png>

Le traitement d'AsciiDoc

Asciidoctor lit et analyse la syntaxe du texte écrit en AsciiDoc afin de créer une représentation, sous forme d'arbre, à partir de laquelle des templates sont appliqués pour produire de l'HTML5, du DocBook 5 et des pages de man(uel).

Vous avez la possibilité d'écrire votre propre convertisseur ou de fournir des templates supportant **Tilt** pour personnaliser le résultat généré ou pour produire des formats alternatifs.

Asciidoctor remplace le processeur AsciiDoc original écrit en Python (`asciidoc.py`). La suite de tests Asciidoctor possède **plus de 2,000 tests** afin de garantir la compatibilité avec la syntaxe AsciiDoc.

En plus de la syntaxe AsciiDoc standard, Asciidoctor reconnaît des balises additionnelles ainsi que des options de formatage, comme les polices d'icônes (par exemple `icon:fire[]`) et des éléments d'interface (par exemple `button:[Enregistrer]`). Asciidoctor offre aussi un thème moderne et « responsive » basé sur **Foundation** pour styliser le document HTML5 généré.

Asciidoctor est disponible partout où Ruby est disponible

Vous pouvez exécuter Asciidoctor dans la JVM en utilisant JRuby. Pour invoquer l'API Asciidoctor directement depuis Java ou d'autres langages de la JVM, utilisez [AsciidoctorJ](#). Des plugins basés sur AsciidoctorJ permettent d'intégrer le processeur Asciidoctor avec Apache Maven, Gradle ou Javadoc.

Asciidoctor s'exécute également au sein de JavaScript. Nous utilisons [Opal](#) pour transcrire le code source Ruby en JavaScript afin de produire [Asciidoctor.js](#), une version pleinement fonctionnelle d'Asciidoctor qui s'intègre dans tout environnement JavaScript, comme un navigateur web ou Node.js. Asciidoctor.js est utilisé pour faire fonctionner les extensions AsciiDoc Preview pour Chrome, Atom, Brackets et autres outils web.

Prérequis

Asciidoctor fonctionne sur Linux, macOS et Windows et requiert une des implémentations suivantes de [Ruby](#) :

- Ruby (MRI/CRuby 1.8.7 - 2.5)
- JRuby (1.7 en mode Ruby 1.8 et 1.9, 9000)
- Rubinius 2.2.x
- Opal (JavaScript)

Si vous utilisez un environnement Windows dans une autre langue que l'anglais, vous pourriez tomber sur l'erreur `Encoding::UndefinedConversionError` lors du lancement d'Asciidoctor. Pour corriger ce problème, nous recommandons de changer la page de code en UTF-8 dans votre console :



```
chcp 65001
```

Après ce changement, tous les maux de tête liés à l'Unicode seront derrière vous. Si vous utilisez un environnement de développement comme Eclipse, assurez-vous de définir l'encodage en UTF-8. Asciidoctor fonctionne mieux lorsque vous utilisez UTF-8 partout.

Installation

Asciidoctor peut être installé en utilisant (a) un gestionnaire de paquets Linux, (b) Homebrew pour macOS, (c) la commande `gem install` (recommandé pour les utilisateurs Windows), (d) l'image officielle Docker, ou (e) Bundler.

L'avantage d'utiliser le gestionnaire de paquets pour installer la gemme est que l'installation englobe celle des librairies Ruby et RubyGems si elles ne sont pas déjà installées.

(a) Gestionnaires de paquets Linux

La version installée par votre gestionnaire de paquets peut ne pas correspondre à la dernière version d'Asciidoctor. Consulter le dépôt de votre distribution Linux pour connaître la dernière version disponible d'Asciidoctor en fonction de la version de votre distribution.

- [Alpine Linux \(asciidoctor\)](#)
- [Debian \(asciidoctor\)](#)
- [Fedora \(asciidoctor\)](#)
- [OpenSUSE \(rubygem-asciidoctor\)](#)
- [Ubuntu \(asciidoctor\)](#)

Si vous souhaitez installer une version plus récente d'Asciidoctor que celle proposée par votre gestionnaire de paquets, suivre [la procédure d'installation gem](#).

apk (Alpine Linux)

Pour installer le paquet sur Alpine Linux, ouvrez un terminal et tapez :

```
$ sudo apk add asciidoctor
```

APT

Sur Debian et les distributions dérivées de Debian, comme Ubuntu, utilisez APT pour

installer Asciidoctor. Pour installer le paquet, ouvrez un terminal et tapez :

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

DNF

Sur les distributions Linux qui utilisent des RPM, comme Fedora, CentOS, et RHEL, utilisez le gestionnaire de paquets DNF pour installer Asciidoctor. Pour installer le paquet, ouvrez un terminal et tapez :

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

(b) Homebrew (macOS)

Vous pouvez utiliser Homebrew, le gestionnaire de paquets sur macOS, pour installer Asciidoctor. Si vous n'avez pas encore installé Homebrew, suivez les instructions sur brew.sh.

Une fois Homebrew installé, vous pouvez installer Asciidoctor. Ouvrez un terminal et tapez :

```
$ brew install asciidoctor
```

Homebrew installe la gemme `asciidoctor` dans un répertoire spécifique qui est indépendant des gemmes système.

(c) gem install

Avant d'installer Asciidoctor en utilisant `gem install`, il est recommandé d'utiliser [RVM](#) pour installer Ruby dans votre « home » (c'est-à-dire, votre espace utilisateur). Ensuite, vous pouvez utiliser la commande `gem` pour installer ou mettre à jour la gemme Asciidoctor. Quand vous utilisez RVM, les gemmes sont installées dans un répertoire isolé du système.

Ouvrez un terminal et tapez :

```
$ gem install asciidoctor
```

Si vous souhaitez installer une version pre-release (c'est-à-dire, une « release candidate »), utilisez :

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

(d) Docker

Lire [Installer Asciidoctor en utilisant Docker](#).

(e) Bundler

1. Créez un fichier Gemfile à la racine de votre projet (ou du répertoire courant)
2. Ajoutez la gemme `asciidoctor` dans votre fichier Gemfile comme ci-dessous :

```
source 'https://rubygems.org'
gem 'asciidoctor'
# ou spécifier la version explicitement
# gem 'asciidoctor', '1.5.7'
```

3. Sauvegardez le fichier Gemfile
4. Ouvrez un terminal et installez la gemme en utilisant :

```
$ bundle
```

Pour mettre à jour la gemme, spécifiez la nouvelle version dans le fichier Gemfile et exécutez `bundle` à nouveau. Utiliser `bundle update` **n'est pas** recommandé car les autres gemmes seront également mises à jour, ce qui n'est pas forcément le résultat voulu.

Mise à jour

Si vous avez installé Asciidoctor en utilisant votre gestionnaire de paquets, votre système d'exploitation est sûrement configuré pour mettre à jour automatiquement les paquets, si tel est le cas vous n'avez pas besoin de mettre à jour manuellement Asciidoctor.

apk (Alpine Linux)

Pour mettre à jour Asciidoctor, tapez :

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

APT

Pour mettre à jour Asciidoctor, tapez :

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

DNF

Pour mettre à jour Asciidoctor, tapez :

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

Homebrew (macOS)

Pour mettre à jour Asciidoctor, tapez :

```
$ brew update  
$ brew upgrade asciidoctor
```

gem install

Si vous avez précédemment installé Asciidoctor en utilisant la commande `gem`, vous devez manuellement mettre à jour Asciidoctor quand une nouvelle version est publiée. Vous pouvez mettre à jour Asciidoctor en tapant :

```
$ gem install asciidoctor
```

Quand vous installez une nouvelle version en utilisant `gem install`, vous vous retrouvez avec plusieurs versions installées. Utilisez la commande ci-dessous pour supprimer les anciennes versions :

```
$ gem cleanup asciidoctor
```

Utilisation

Si la gemme AsciiDoctor s’est installée correctement, la ligne de commande (CLI) `asciidoctor` sera disponible dans votre PATH. Pour vérifier sa disponibilité, exécutez la commande suivante dans votre terminal :

```
$ asciidoctor --version
```

Vous devriez voir les informations concernant la version d’AsciiDoctor et celle de votre environnement Ruby s’afficher dans le terminal.

```
AsciiDoctor 1.5.7 [http://asciidoctor.org]  
Runtime Environment (ruby 2.4.1p111 [x86_64-linux]) (lc:UTF-8 fs:UTF-8 in:- ex:UTF-8)
```

AsciiDoctor fournit aussi une API. Cette API permet une intégration avec d’autres logiciels Ruby, comme Rails, Sinatra et GitHub, ainsi que d’autres langages comme Java (via [AsciiDoctorJ](#)) ou JavaScript (via [AsciiDoctor.js](#)).

Interface de Ligne de Commande (CLI)

La commande `asciidoctor` vous permet d’invoquer AsciiDoctor à partir de la ligne de commande (c’est-à-dire, un terminal).

La commande suivante convertit le fichier README.adoc en HTML et sauvegarde le résultat dans le fichier README.html dans le même répertoire. Le nom du fichier HTML généré est tiré de celui du fichier source, l’extension a été changée pour

.html.

```
$ asciidoctor README.adoc
```

Vous pouvez contrôler le processeur Asciidoctor en ajoutant plusieurs paramètres, vous pouvez en apprendre plus sur ces derniers en utilisant la commande :

```
$ asciidoctor --help
```

Par exemple, pour écrire le fichier dans un répertoire différent, utilisez :

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

La [page man](#) `asciidoctor` fournit une référence complète sur l'interface de ligne de commande.

Référez-vous aux ressources suivantes pour en apprendre davantage sur la façon d'utiliser la commande `asciidoctor`.

- [Comment convertir un document ?](#)
- [Comment utiliser la fabrique de feuilles de style Asciidoctor pour produire des thèmes personnalisés ?](#)

API Ruby

Pour utiliser Asciidoctor dans votre application, vous avez tout d'abord besoin de faire un « require » sur la gemme :

```
require 'asciidoctor'
```

Vous pouvez ensuite convertir un fichier AsciiDoc en fichier HTML en utilisant :

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



Quand vous utilisez Asciidoctor via l'API, le mode de sûreté par défaut est `:secure`. Dans le mode « secure », plusieurs fonctionnalités centrales sont désactivées, comme la directive `include`. Si vous souhaitez activer ces fonctionnalités, vous aurez besoin de définir explicitement le mode de sûreté avec une la valeur `:server` (recommandée) ou `:safe`.

Vous pouvez aussi convertir une chaîne de texte en fragment HTML (pour une insertion dans une page HTML) en utilisant :

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

Si vous voulez le document HTML complet, activez l'option `header_footer` comme ci-dessous :

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

Si vous avez besoin d'accéder au document analysé, vous pouvez séparer la conversion en deux étapes distinctes :

```
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
```

Gardez en tête que si vous n'aimez pas le contenu généré par Asciidoctor, *vous pouvez le changer* ! Asciidoctor supporte des convertisseurs personnalisés qui peuvent prendre en charge la conversion depuis le document analysé jusqu'au contenu généré.

Une façon simple de personnaliser les morceaux de contenu générés est d'utiliser le convertisseur de template. Le convertisseur de template vous permet, en utilisant un template supporté par [Tilt](#), de prendre en charge la conversion de n'importe quel élément dans le document.

Vous l'aurez compris, vous *pouvez* complètement prendre le contrôle sur le contenu généré. Pour plus d'informations sur comment utiliser l'API ou personnaliser le contenu généré, référez-vous au [manuel utilisateur](#).

Contributions

Les contributeurs et contributrices sont toujours les bienvenus ! Si vous découvrez des erreurs ou des oublis dans le code source, la documentation, ou le contenu du site web, s'il vous plaît n'hésitez pas à ouvrir un ticket ou une « pull request » avec un correctif.

Voici quelques façons de contribuer :

- en utilisant les versions prerelease (alpha, beta ou preview),
- en rapportant des anomalies,
- en suggérant de nouvelles fonctionnalités,
- en écrivant ou éditant la documentation,
- en écrivant du code avec des tests — *Aucun patch n'est trop petit*
 - corriger une coquille,
 - ajouter des commentaires,
 - nettoyer des espaces inutiles,
 - écrire des tests !
- en refactorant le code,
- en corrigeant des [anomalies](#),
- en effectuant des relectures des patches.

Le guide du [parfait Contributeur](#) fournit des informations sur comment créer, styliser et soumettre des tickets, des demandes de fonctionnalités, du code et de la documentation pour le projet AsciiDoctor.

Être aidé

Asciidoctor est développé dans le but de vous aider à écrire et publier du contenu. Mais nous ne pouvons pas le faire sans vos avis ! Nous vous encourageons à poser vos questions et à discuter de n'importe quels aspects du projet sur la liste de discussion, Twitter ou dans le salon de discussion.

Chat (Gitter)

[\[Gitter\]](#)

Forum (Nabble)

<http://discuss.asciidoctor.org>

Twitter

hashtag [#asciidoctor](#) ou la mention [@asciidoctor](#)

L'organisation Asciidoctor sur GitHub héberge le code source du projet, le gestionnaire de tickets ainsi que des sous-projets.

Dépôt des sources (git)

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

Gestionnaire de tickets

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

L'organisation AsciiDoctor sur GitHub

<https://github.com/asciidoctor>

Licence

Copyright © 2012-2018 Dan Allen, Sarah White et les contributeurs individuels d'AsciiDoctor. Une utilisation libre de ce logiciel est autorisée sous les termes de la licence MIT.

Consultez le fichier [LICENSE](#) pour plus de détails.

Auteurs

AsciiDoctor est mené par [Dan Allen](#) et [Sarah White](#) et reçoit de nombreuses contributions de la part de la [géniale communauté](#) AsciiDoctor. Le projet a été initié en 2012 par [Ryan Waldron](#) et est basé sur [un prototype](#) écrit par [Nick Hengeveld](#).

AsciiDoc a été démarré par Stuart Rackham et a reçu de nombreuses contributions de la part de la communauté AsciiDoc.

Changelog

1.5.7.1 (2018-05-10) - @mojomlinux

Fixes

- fix regression where block attributes were being inherited by sibling blocks in a

complex list item (#2771)

- don't apply lead styling to first paragraph in nested document (AsciiDoc table cell) if role is present (#2624)

Build / Infrastructure

- drop obsolete logic in rake build (@aerostitch)
- allow lib dir to be overridden for tests using an environment variable (PR #2758) (@aerostitch)
- load asciidoctor/version from LOAD_PATH in gemspec if not found locally (PR #2760) (@aerostitch)

1.5.7 (2018-05-02) - @mojavelinux

Enhancements

- BREAKING: drop XML tags, character refs, and non-word characters (except hyphen, dot, and space) when auto-generating section IDs (#794)
 - hyphen, dot, and space are replaced with value of idseparator, if set; otherwise, spaces are dropped
- allow attribute names to contain any word character defined by Unicode (#2376, PR #2393)

- do not recognize attribute entry line if name contains a colon (PR #2377)
- route all processor messages through a logger instead of using Kernel#warn (#44, PR #2660)
- add MemoryLogger for capturing messages sent to logger into memory (#44, PR #2660)
- add NullLogger to prevent messages from being logged (#44, PR #2660)
- log message containing source location / cursor as an object; provides more context (#44, PR #2660)
- pass cursor for include file to :include_location key in message context (PR #2729)
- add :logger option to API to set logger instance (#44, PR #2660)
- add --failure-level=LEVEL option to CLI to force non-zero exit code if specified logging level is reached (#2003, PR #2674)
- parse text of xref macro as attributes if attribute signature found (equal sign) (#2381)
- allow xrefstyle to be specified per xref by assigning the xrefstyle attribute on the xref macro (#2365)

- recognize target with .adoc extension in xref macro as an interdocument xref
- resolve nested includes in remote documents relative to URI (#2506, PR #2511)
- allow `relfilesuffix` attribute to control file extension used for interdoc xrefs (#1273)
- support `!name@` (preferred), `!name=@`, `name!@`, and `name!=@` syntax to soft unset attribute from API or CLI (#642, PR #2649)
- allow modifier to be placed at end of name to soft set an attribute (e.g., `icons@=font`) (#642, PR #2649)
- interpret `false` attribute value defined using API as a soft unset (#642, PR #2649)
- number parts if `partnums` attribute is set (#2298)
- allow footnote macro to define or reference footnote reference (footnoteref macro now deprecated) (#2347, PR #2362)
- allow custom converter to be used with custom templates; converter must declare that it supports templates (#2619)
- allow manpage path for manpage help topic to be specified using `ASCIIDOCTOR_MANPAGE_PATH` environment variable (PR #2653)

(**@aerostitch**)

- if manpage cannot be found in default path inside gem, use `man -w asciidoctor` to resolve installed path (PR #2653)
- uncompress contents of manpage for manpage help topic if path ends with `.gz` (PR #2653) (**@aerostitch**)
- define source and manual refmiscinfo entries in manpage output if manual and source attributes are defined (PR #2636) (**@tiwai**)
- add syntax for adding hard line breaks in block AsciiMath equations (#2497, PR #2579) (**@dimztimz**)
- add positioning option to sectanchors attribute (sectanchors=before or sectanchors=after) (#2485, PR #2486)
- allow table striping to be configured using stripes attribute (even, odd, all, or none) or stripes roles on table (#1365, PR #2588)
- recognize ends as an alias to topbot for configuring the table frame
- add rel=nofollow property to links (text or image) when nofollow option is set (#2605, PR #2692)
- populate Document#source_location when sourcemap option is enabled (#2478, PR #2488)

- populate source_location property on list items when sourcemap option is set on document (PR #2069) (@mogztter)
- populate Table::Cell#source_location when sourcemap option is enabled (#2705)
- allow local include to be flagged as optional by setting optional option (#2389, PR #2413)
- allow block title to begin with a period (#2358, PR #2359)
- catalog inline anchor at start of list items in ordered and unordered lists, description list terms, and table cells (#2257)
- register document in catalog if id is set; assign ref text to document attributes if specified in a block attribute line (#2301, PR #2428)
- allow automatic width to be applied to individual columns in a table using the special value ~ (#1844)
- use the quote element in DocBook converter to represent smart quotes (#2272, PR #2356) (@bk2204)
- parse and pass all manpage names to output document master (i.e., shadow man pages) (#1811, #2543, PR #2414)

- parse credit line of shorthand quote block as block attributes; apply normal subs to credit line in shorthand quote blocks (#1667, PR #2452)
- populate copyright element in DocBook output from value of copyright attribute (#2728)
- preserve directories if source dir and destination dir are set (#1394, PR #2421)
- allow linkcss to be unset from API or CLI when safe mode is secure
- convert quote to epigraph element in DocBook output if block has epigraph role (#1195, PR #2664) (@**bk2204**)
- number special sections in addition to regular sections when sectnums=all (#661, PR #2463)
- upgrade to Font Awesome 4.7.0 (#2569)
- upgrade to MathJax 4.7.4

Bug fixes

- set :to_dir option value correctly when output file is specified (#2382)
- preserve leading indentation in contents of AsciiDoc table cell if contents starts with a newline (#2712)
- the shorthand syntax on the style to set block attributes (id, roles, options) no

longer resets block style (#2174)

- match include tags anywhere on line as long as offset by word boundary on left and space or newline on right (#2369, PR #2683)
- warn if an include tag specified in the include directive is unclosed in the included file (#2361, PR #2696)
- use correct parse mode when parsing blocks attached to list item (#1926)
- fix typo in gemspec that removed README and CONTRIBUTING files from the generated gem (PR #2650) (@aerostitch)
- preserve id, role, title, and reftext on open block when converting to DocBook; wrap in <para> or <formalpara> (#2276)
- don't turn bare URI scheme (no host) into a link (#2609, PR #2611)
- don't convert inter-document xref to internal anchor unless entire target file is included into current file (#2200)
- fix em dash replacement in manpage converter (#2604, PR #2607)
- don't output e-mail address twice when replacing bare e-mail address in manpage output (#2654, PR #2665)
- use alternate macro for monospaced text in manpage output to not conflict w/ AsciiDoc macros (#2751)

- enforce that absolute start path passed to PathResolver#system_path is inside of jail path (#2642, PR #2644)
- fix behavior of PathResolver#descends_from? when base path equals / (#2642, PR #2644)
- automatically recover if start path passed to PathResolver#system_path is outside of jail path (#2642, PR #2644)
- re-enable left justification after invoking tmac URL macro (#2400, PR #2409)
- don't report warning about same level 0 section multiple times (#2572)
- record timings when calling convert and write on Document (#2574, PR #2575)
- duplicate header attributes when restoring; allows header attributes to be restored an arbitrary number of times (#2567, PR #2570)
- propagate :catalog_assets option to nested document (#2564, PR #2565)
- preserve newlines in quoted CSV data (#2041)
- allow opening quote around quoted CSV field to be on a line by itself
- output table footer after body rows (#2556, PR #2566) (**@PauloFrancaLacerda**)
- move @page outside of @media print in default stylesheet (#2531, PR #2532)

- don't throw exception if text of dd node is nil (#2529, PR #2530)
- don't double escape ampersand in manpage output (#2525) (@**dimztimz**)
- fix crash when author_1 attribute is assigned directly (#2481, PR #2487)
- fix CSS for highlighted source block inside colist (#2474, PR #2490)
- don't append file extension to data uri of admonition icon (#2465, PR #2466)
- fix race condition in Helpers.mkdir_p (#2457, PR #2458)
- correctly process nested passthrough inside unconstrained monospaced (#2442, PR #2443)
- add test to ensure ampersand in author line is not double escaped (#2439, PR #2440)
- prevent footnote ID from clashing with auto-generated footnote IDs (#2019)
- fix alignment of icons in footnote (#2415, PR #2416)
- add graceful fallback if pygments.rb fails to return a value (#2341, PR #2342)
- escape specialchars in source if pygments fails to highlight (#2341)
- do not recognize attribute entry line if name contains colon (PR #2377)
- allow flow indexterm to be enclosed in round brackets (#2363, PR #2364)

- set outfilesuffix to match file extension of output file (#2258, PR #2367)
- add block title to dlist in manpage output (#1611, PR #2434)
- scale text to 80% in print styles (#1484, PR #2576)
- fix alignment of abstract title when using default stylesheet (PR #2732)
- only set nowrap style on table caption for auto-width table (#2392)
- output non-breaking space for man manual if absent in DocBook output (PR #2636)
- don't crash if stem type is not recognized (instead, fallback to asciimath)

Improvements / Refactoring

- BREAKING: rename table spread role to stretch (#2589, PR #2591)
- use cursor marks to track lines more accurately; record cursor at the start of each block, list item, or table cell (PR #2701, PR #2547) (@seikichi)
- log a warning message if an unterminated delimited block is detected (#1133, PR #2612)
- log a warning when nested section is found inside special section that doesn't support nested sections (#2433, PR #2672)
- read files in binary mode to disable automatic endline coercion (then explicitly

coerce to UTF-8) (PR #2583, PR #2694)

- resolve / expand parent references in start path passed to PathResolver#system_path (#2642, PR #2644)
- update PathResolver#expand_path to resolve parent references (#2642, PR #2644)
- allow start path passed to PathResolver#system_path to be outside jail if target brings resolved path back inside jail (#2642, PR #2644)
- don't run File.expand_path on Dir.pwd (assume Dir.pwd is absolute) (#2642, PR #2644)
- posixify working_dir passed to PathResolver constructor if absolute (#2642, PR #2644)
- optimize detection for footnote* and indexterm* macros (#2347, PR #2362)
- log a warning if a footnote reference cannot be resolved (#2669)
- set logger level to DEBUG when verbose is enabled
- coerce value of :template_dirs option to an Array (PR #2621)
- make block roles specified using shorthand syntax additive (#2174)
- allow paragraph to masquerade as open block (PR #2412)

- move callouts into document catalog (PR #2394)
- document ID defined in block attribute line takes precedence over ID defined inside document title line
- don't look for link and window attributes on document when resolving these attributes for an image
- when linkattrs is set, only parse attributes in link macro if equals is present
- skip line comments in name section of manpage (#2584, PR #2585)
- always activate extension registry passed to processor (PR #2379)
- skip extension registry activation if no groups are registered (PR #2373)
- don't apply lead styling to first paragraph if role is present (#2624, PR #2625)
- raise clearer exception when extension class cannot be resolved (#2622, PR #2623)
- add methods to read results from timings (#2578, PR #2580)
- collapse bottom margin of last block in AsciiDoc table cell (#2568, PR #2593)
- set authorcount to 0 if there are no authors (#2519, PR #2520)
- validate fragment of interdoc xref that resolves to current doc (#2448, PR #2449)

- put id attribute on tag around phrase instead of preceding anchor (#2445, PR #2446)
- add .plist extension to XML circumfix comment family (#2430, PR #2431) (@akosma)
- alias Document#title method to no args Document#doctitle method (#2429, PR #2432)
- upgrade missing or unreadable include file to an error (#2424, PR #2426)
- add compliance setting to disable natural cross references (#2405, PR #2460)
- make hash in inter-document xref target optional if target has extension (#2404, PR #2406)
- add CSS class to part that matches role (#2401, PR #2402)
- add fit-content class to auto-width table (#2392)
- automatically assign parent reference when adding node to parent (#2398, PR #2403)
- leave inline anchor in section title as is if section has ID (#2243, PR #2427)
- align and improve error message about invalid use of partintro between HTML5 and DocBook converters

- rephrase warning when level 0 sections are found and the doctype is not book
- report correct line number when duplicate bibliography anchor is found
- only warn if thread_safe gem is missing when using built-in template cache
- rename enumerate_section to assign_numeral; update API docs
- drop deprecated compact option from CLI; remove from manpage
- use more robust mechanism for lazy loading the asciimath gem
- use consistent phrase to indicate the processor is automatically recovering from a problem
- change Reader#skip_comment_lines to not return skipped lines
- add styles to default stylesheet for display on Kindle (kf8) devices (PR #2475)
- purge render method from test suite (except to verify alias)

Documentation

- translate 'section-refsig' for German language (PR #2633) (@**ahus1**)
- synchronize French README with English version (PR #2637) (@**flashcode**)

Build / Infrastructure

- create an official logo for the project (#48) (@**mmajko**)

- update Ruby versions in appveyor build matrix (PR #2388) (@**miltador**)
- add mailinglist, changelog, source, and issues URI to gem spec
- allow blocks and substitutions tests to be run directly
- asciidoctor formula now available for Homebrew (@**zmxwangx**)

Distribution Packages

- [RubyGem \(asciidoctor\)](#)
- [Fedora \(asciidoctor\)](#)
- [Debian \(asciidoctor\)](#)
- [Ubuntu \(asciidoctor\)](#)
- [Alpine Linux \(asciidoctor\)](#)
- [OpenSUSE \(rubygem-asciidoctor\)](#)

Référez-vous au fichier [CHANGELOG](#) pour une liste complète des changements des versions précédentes.