AsciiDoc Syntax Quick Reference

Table of Contents

AsciiDoc is a lightweight markup language for authoring notes, articles, documentation, books, web pages, slide decks and man pages in plain text. This guide is a quick reference for the common AsciiDoc document and text formatting markup.

> **NOTE** These examples focus on the output generated by the HTML backend. AsciiDoc produces complementary output when generating PDF, EPUB, and DocBook.

Paragraphs

Normal

```
Paragraphs don't require any special markup in AsciiDoc.
A paragraph is just one or more lines of consecutive text.

To begin a new paragraph, separate it by at least one blank line.
Newlines within a paragraph are not displayed.
```

Paragraphs don't require any special markup in AsciiDoc. A paragraph is just one or more lines of consecutive text.

To begin a new paragraph, separate it by at least one blank line. Newlines within a paragraph are not displayed.

Literal

```
A normal paragraph.

 A paragraph offset by at least one space becomes a literal paragraph.
 All lines in a literal paragraph must be adjacent.

 A literal paragraph is displayed as preformatted text.
 The text is shown in a fixed-width font.
 Spaces and newlines,
 like the ones in this sentence,
 are preserved.

Another normal paragraph.
```

A normal paragraph.

```
 A paragraph offset by at least one space becomes a literal paragraph.
 All lines in a literal paragraph must be adjacent.
```

```
 A literal paragraph is displayed as preformatted text.
 The text is shown in a fixed-width font.
 Spaces and newlines,
 like the ones in this sentence,
 are preserved.
```

Another normal paragraph.

Admonition

```
NOTE: An admonition paragraph draws the reader's attention to
auxiliary information.
Its purpose is determined by the label
at the beginning of the paragraph.

Here are the other built-in admonition types:

TIP: Pro tip...

IMPORTANT: Don't forget...

WARNING: Watch out for...

CAUTION: Ensure that...
```

NOTE An admonition paragraph draws the reader's attention to auxiliary information. Its purpose is determined by the label at the beginning of the paragraph.

Here are the other built-in admonition types:

TIP Pro tip...

IMPORTANT Don't forget...

WARNING Watch out for...

CAUTION Ensure that...

NOTE You can also create admonition blocks.

Lead paragraph

```
[.lead]
This text will be styled as a lead paragraph (i.e., larger font).
```

This text will be styled as a lead paragraph (i.e., larger font).

NOTE The default Asciidoctor stylesheet automatically styles the first paragraph of the preamble as a lead paragraph.

> **More Paragraph, Admonition and Literal Block Examples**
>
> See these sections in the Asciidoctor User Manual for more information and examples.
>
> - Paragraphs
> - Literal Text and Blocks
> - Admonitions

Formatted Text

Bold, Italic, and Monospace

```
bold *constrained* & **un**constrained

italic _constrained_ & __un__constrained

bold italic *_constrained_* & **__un__**constrained

monospace `constrained` & ``un``constrained

monospace bold `*constrained*` & ``**un**``constrained

monospace italic `_constrained_` & ``__un__``constrained

monospace bold italic `*_constrained_*` & ``**__un__**``constrained
```

bold **constrained** & **un**constrained

italic *constrained* & *un*constrained

bold italic ***constrained*** & ***un***constrained

monospace `constrained` & `un`constrained

monospace bold **`constrained`** & **`un`**constrained

monospace italic *`constrained`* & *`un`*constrained

monospace bold italic ***`constrained`*** & ***`un`***constrained

Monospace vs codespan

```
`{cpp}` is valid syntax in the programming language by the same name.

`+WHERE id <= 20 AND value = "{name}"+` is a SQL WHERE clause.
```

C++ is valid syntax in the programming language by the same name.

WHERE id <= 20 AND value = "{name}" is a SQL WHERE clause.

NOTE The meaning of backtick (\`) and plus (+) changed in Asciidoctor 1.5.0. Backticks only make the text monospaced, whereas pluses passthrough text without applying formatting. See the migration page for details.

Marks and Custom Styling

```
Werewolves are allergic to #cassia cinnamon#.

Did the werewolves read the [.small]#small print#?

Where did all the [.underline]#cores# run off to?

We need [.line-through]#ten# make that twenty VMs.

[.big]##O##nce upon an infinite loop.
```

Werewolves are allergic to cassia cinnamon.

Did the werewolves read the small print?

Where did all the cores run off to?

We need ten make that twenty VMs.

Once upon an infinite loop.

Superscript and Subscript

```
^super^script phrase

~sub~script phrase
```

super script phrase

sub script phrase

Curved Quotation Marks and Apostrophes (Smart Quotes)

```
"`double curved quotes`"

'`single curved quotes`'

Olaf's desk was a mess.

All of the werewolves`' desks were a mess.

Olaf had been with the company since the `'60s.
```

"double curved quotes"

'single curved quotes'

Olaf's desk was a mess.

All of the werewolves' desks were a mess.

Olaf had been with the company since the '60s.

More Text Formatting Examples

See these sections in the Asciidoctor User Manual for more information and examples.

- Bold and Italic Formatting
- Quotation Marks and Apostrophes
- Subscript and Superscript
- Monospace Formatting
- Custom Styling with Attributes
- Passthrough Macros

Document Header

IMPORTANT | A header is optional.

CAUTION | The header may not contain blank lines and must be offset from the content by at least one blank line.

Title only

```
= My Document's Title

My document provides...
```

Title and author line

```
= My Document's Title
Doc Writer <doc.writer@asciidoctor.org>

My document provides...
```

> **TIP** Asciidoctor allows multiple authors in the author line. Use the semi-colon character to separate each author.

Title, author line and revision line

```
= My Document's Title
Doc Writer <doc.writer@asciidoctor.org>
v1.0, 2014-01-01

My document provides...
```

> **IMPORTANT** You cannot have a revision line without an author line.

Document header with attributes

```
= My Document's Title
Doc Writer <doc.writer@asciidoctor.org>
v1.0, 2018-04-11
:toc:
:imagesdir: assets/images
:homepage: https://asciidoctor.org

My document provides...
```

Section Titles (Headings)

```
= Document Title (Level 0)

== Level 1 Section Title

=== Level 2 Section Title

==== Level 3 Section Title

===== Level 4 Section Title

====== Level 5 Section Title

== Another Level 1 Section Title
```

Document Title (Level 0)

Level 1 Section Title

Level 2 Section Title

Level 3 Section Title

Level 4 Section Title

Level 5 Section Title

Another Level 1 Section Title

WARNING When using the article doctype (the default), you can only have one level-0 section title (i.e., the document title) and it must be in the document header.

NOTE The number of equal signs matches the heading level in the HTML output. For example, *Section Level 1* becomes an <h2> heading.

```
= Document Title (Level 0)

== Section Level 1

=== Section Level 2

==== Section Level 3

===== Section Level 4

====== Section Level 5

= Section Level 0
```

Document Title (Level 0)

Section Level 1

Section Level 2

Section Level 3

Section Level 4

Section Level 5

Section Level 0

Explicit id

```
[#primitives-nulls]
== Primitive types and null values
```

Section anchors and links (Asciidoctor only)

sectanchors

When this document attribute is set, a section icon anchor appears in front of the section title.

sectlinks

When this document attribute is set, the section titles become self-links. This enables a reader to bookmark the section.

NOTE	Section title anchors depend on the default Asciidoctor stylesheet to render properly.

Include Files

Document parts

```
= Reference Documentation
Lead Developer

This is documentation for project X.

include::basics.adoc[]

include::installation.adoc[]

include::example.adoc[]
```

CAUTION	Asciidoctor does not insert blank lines between adjacent include statements to keep the content separated. Be sure to add a blank line in the source document to avoid unexpected results, such as a section title being swallowed.

Include content from a URI

```
include::https://raw.githubusercontent.com/asciidoctor/asciidoctor/master/README.adoc[
]
```

NOTE	Including content from a URI is potentially dangerous, so it's disabled if the safe mode is SECURE or greater. Assuming the safe mode is less than SECURE, you must also set the `allow-uri-read` attribute to permit Asciidoctor to read content from a URI.

Breaks

Hard line break

```
Rubies are red, +
Topazes are blue.

[%hardbreaks]
Ruby is red.
Java is black.
```

> Rubies are red,
> Topazes are blue.
>
> Ruby is red.
> Java is black.

Thematic break (aka horizontal rule)

```
before

'''

after
```

before

---

after

Page break

```
<<<
```

Lists

Unordered, basic

```
* Edgar Allen Poe
* Sheri S. Tepper
* Bill Bryson
```

- Edgar Allen Poe

- Sheri S. Tepper

- Bill Bryson

Unordered, basic (alt)

```
- Edgar Allen Poe
- Sheri S. Tepper
- Bill Bryson
```

- Edgar Allen Poe
- Sheri S. Tepper
- Bill Bryson

Unordered, max nesting

```
* level 1
** level 2
*** level 3
**** level 4
***** level 5
* level 1
```

- level 1
  - level 2
    - level 3
      - level 4
        - level 5
- level 1

TIP | The unordered list marker can be changed using block styles.

Ordered, basic

```
. Step 1
. Step 2
. Step 3
```

1. Step 1
2. Step 2
3. Step 3

Ordered, nested

```
. Step 1
. Step 2
.. Step 2a
.. Step 2b
. Step 3
```

1. Step 1

2. Step 2

    a. Step 2a

    b. Step 2b

3. Step 3

Ordered, max nesting

```
. level 1
.. level 2
... level 3
.... level 4
..... level 5
. level 1
```

1. level 1

    a. level 2

        i. level 3

            A. level 4

                I. level 5

2. level 1

Checklist

```
* [*] checked
* [x] also checked
* [ ] not checked
*     normal list item
```

☑ checked

☑ also checked

☐ not checked

normal list item

> **TIP**   Checklists can use font-based icons and be interactive.

Description, single-line

```
first term:: definition of first term
second term:: definition of second term
```

first term

definition of first term

second term

definition of second term

Description, multi-line

```
first term::
definition of first term
second term::
definition of second term
```

first term

definition of first term

second term

definition of second term

Q&A

```
[qanda]
What is Asciidoctor?::
  An implementation of the AsciiDoc processor in Ruby.
What is the answer to the Ultimate Question?:: 42
```

1. *What is Asciidoctor?*

   An implementation of the AsciiDoc processor in Ruby.

2. *What is the answer to the Ultimate Question?*

   42

Mixed

```
Operating Systems::
  Linux:::
    . Fedora
      * Desktop
    . Ubuntu
      * Desktop
      * Server
  BSD:::
    . FreeBSD
    . NetBSD

Cloud Providers::
  PaaS:::
    . OpenShift
    . CloudBees
  IaaS:::
    . Amazon EC2
    . Rackspace
```

Operating Systems

Linux

1. Fedora
   - Desktop
2. Ubuntu
   - Desktop
   - Server

BSD

1. FreeBSD
2. NetBSD

Cloud Providers

PaaS

1. OpenShift
2. CloudBees

IaaS

1. Amazon EC2
2. Rackspace

TIP	Lists can be indented. Leading whitespace is not significant.

Complex content in outline lists

```
* Every list item has at least one paragraph of content,
  which may be wrapped, even using a hanging indent.
+
Additional paragraphs or blocks are adjoined by putting
a list continuation on a line adjacent to both blocks.
+
list continuation:: a plus sign (`{plus}`) on a line by itself

* A literal paragraph does not require a list continuation.

 $ gem install asciidoctor

* AsciiDoc lists may contain any complex content.
+
[cols="2", options="header"]
|===
|Application
|Language

|AsciiDoc
|Python

|Asciidoctor
|Ruby
|===
```

- Every list item has at least one paragraph of content, which may be wrapped, even using a hanging indent.

  Additional paragraphs or blocks are adjoined by putting a list continuation on a line adjacent to both blocks.

  list continuation

  > a plus sign (+) on a line by itself

- A literal paragraph does not require a list continuation.

  ```
   $ gem install asciidoctor
  ```

- AsciiDoc lists may contain any complex content.

Application	Language
AsciiDoc	Python
Asciidoctor	Ruby

Links

External

```
https://asciidoctor.org - automatic!

https://asciidoctor.org[Asciidoctor]

https://github.com/asciidoctor[Asciidoctor @ *GitHub*]
```

https://asciidoctor.org - automatic!

[Asciidoctor](https://asciidoctor.org)

[Asciidoctor @ **GitHub**](https://github.com/asciidoctor)

With spaces and special characters

```
link:++https://example.org/?q=[a b]++[URL with special characters]

link:https://example.org/?q=%5Ba%20b%5D[URL with special characters]
```

[URL with special characters](https://example.org/?q=[a%20b])

[URL with special characters](https://example.org/?q=%5Ba%20b%5D)

Windows path

```
link:\\server\share\whitepaper.pdf[Whitepaper]
```

[Whitepaper](\\server\share\whitepaper.pdf)

Relative

```
link:index.html[Docs]
```

[Docs](index.html)

Email and IRC

```
devel@discuss.arquillian.org

mailto:devel@discuss.arquillian.org[Discuss Arquillian]

mailto:devel-join@discuss.arquillian.org[Subscribe,Subscribe me,I want to join!]

irc://irc.freenode.org/#fedora
```

devel@discuss.arquillian.org

Discuss Arquillian

Subscribe

irc://irc.freenode.org/#fedora

Link with attributes (Asciidoctor only)

```
http://discuss.asciidoctor.org[Discuss Asciidoctor,role=external,window=_blank]

http://discuss.asciidoctor.org[Discuss Asciidoctor^]

https://example.org["Google, Yahoo, Bing^",role=teal]
```

Discuss Asciidoctor

Discuss Asciidoctor

Google, Yahoo, Bing

NOTE Links with attributes (including the subject and body segments on mailto links) are a feature unique to Asciidoctor. To enable them prior to 1.5.7, you must set the `linkattrs` attribute on the document. Since 1.5.7, attribute parsing is enabled automatically if an equal sign follows a comma. When attribute parsing is enabled, you must quote the link text if it contains a comma.

Inline anchors

```
[[bookmark-a]]Inline anchors make arbitrary content referenceable.

[#bookmark-b]#Inline anchors can be applied to a phrase like this one.#

anchor:bookmark-c[]Use a cross reference to link to this location.

[[bookmark-d,last paragraph]]The xreflabel attribute will be used as link text in the
cross-reference link.
```

Inline anchors make arbitrary content referenceable.

Inline anchors can be applied to a phrase like this one.

Use a cross reference to link to this location.

The xreflabel attribute will be used as link text in the cross-reference link.

Internal cross references

```
See <<paragraphs>> to learn how to write paragraphs.

Learn how to organize the document into <<section-titles,sections>>.
```

See Paragraphs to learn how to write paragraphs.

Learn how to organize the document into sections.

Inter-document cross references (Asciidoctor only)

```
Refer to <<document-b.adoc#section-b,Section B>> for more information.

See you when you get back from <<document-b#section-b,Section B>>!
```

Images

Images are resolved relative to the value of the imagesdir document attribute, which is empty by default. You are encouraged to make use of the `imagesdir` attribute to avoid hard-coding the common path to your images in every image macro.

The `imagesdir` attribute can be an absolute path, relative path, or base URL. When the image target is a URL or absolute path, the imagesdir prefix is *not* prepended.

Block

```
image::sunset.jpg[]

image::sunset.jpg[Sunset]

.A mountain sunset
[#img-sunset]
[caption="Figure 1: ",link=https://www.flickr.com/photos/javh/5448336655]
image::sunset.jpg[Sunset,300,200]

image::https://asciidoctor.org/images/octocat.jpg[GitHub mascot]
```

Figure 1: A mountain sunset

[GitHub mascot] | *https://asciidoctor.org/images/octocat.jpg*

Inline

```
Click image:icons/play.png[Play, title="Play"] to get the party started.

Click image:icons/pause.png[title="Pause"] when you need a break.
```

Click ▶ to get the party started.

Click ⏸ when you need a break.

IMPORTANT	Two colons following the image keyword in the macro (i.e., `image::`) indicates a block image (aka figure), whereas one colon following the image keyword (i.e., `image:`) indicates an inline image. (All macros follow this pattern). You use an inline image when you need to place the image in a line of text. Otherwise, you should prefer the block form.

Inline image with positioning role

```
image:sunset.jpg[Sunset,150,150,role="right"] What a beautiful sunset!
```


What a beautiful sunset!

TIP	There are a variety of attributes available to position and frame images.

Embedded

```
= Document Title
:data-uri:
```

NOTE	When the `data-uri` attribute is set, all images in the document—including admonition icons—are embedded into the document as data URIs.

TIP	Instead of declaring the `data-uri` attribute in the document, you can pass it as a command-line argument using `-a data-uri`.

Videos

Block

```
video::video_file.mp4[]

video::video_file.mp4[width=640, start=60, end=140, options=autoplay]
```

Embedded Youtube video

```
video::rPQoq7ThGAU[youtube]
```

Embedded Vimeo video

```
video::67480300[vimeo]
```

| TIP | You can control the video settings using additional attributes and options on the macro. |

Source Code

Inline (monospace only)

```
Reference code like `types` or `methods` inline.

Do not pass arbitrary ``Object``'s to methods that accept ``String``'s!
```

Reference code like `types` or `methods` inline.

Do not pass arbitrary `Object`s to methods that accept `String`s!

Inline (literal)

```
Output literal monospace text such as `+{backtick}+` by
enclosing the text in pluses, then in backticks.
```

Output literal monospace text such as `{backtick}` by enclosing the text in pluses, then in backticks.

Literal line

```
 Indent the line one space to insert a code snippet
```

```
Indent the line one space to insert a code snippet
```

Literal block

```
....
error: The requested operation returned error: 1954 Forbidden search for defensive
operations manual
absolutely fatal: operation initiation lost in the dodecahedron of doom
would you like to die again? y/n
....
```

```
error: The requested operation returned error: 1954 Forbidden search for defensive
operations manual
absolutely fatal: operation initiation lost in the dodecahedron of doom
would you like to die again? y/n
```

Listing block with title, no syntax highlighting

```
.Gemfile.lock
----
GEM
  remote: https://rubygems.org/
  specs:
    asciidoctor (1.5.6.1)

PLATFORMS
  ruby

DEPENDENCIES
  asciidoctor (~> 1.5.6.1)
----
```

Gemfile.lock

```
GEM
  remote: https://rubygems.org/
  specs:
    asciidoctor (1.5.6.1)

PLATFORMS
  ruby

DEPENDENCIES
  asciidoctor (~> 1.5.6.1)
```

Code block with title and syntax highlighting

```
.app.rb
[source,ruby]
----
require 'sinatra'

get '/hi' do
  "Hello World!"
end
----
```

app.rb

```
require 'sinatra'

get '/hi' do
  "Hello World!"
end
```

Code block with callouts

```
[source,ruby]
----
require 'sinatra' // <1>

get '/hi' do // <2>
  "Hello World!" // <3>
end
----
<1> Library import
<2> URL mapping
<3> HTTP response body
```

```
require 'sinatra'  ①

get '/hi' do ②
  "Hello World!"  ③
end
```

① Library import

② URL mapping

③ HTTP response body

Code block with non-selectable callouts

```
----
line of code  // <1>
line of code  # <2>
line of code  ;; <3>
----
<1> A callout behind a line comment for C-style languages.
<2> A callout behind a line comment for Ruby, Python, Perl, etc.
<3> A callout behind a line comment for Clojure.
```

```
line of code   ①
line of code   ②
line of code   ③
```

① A callout behind a line comment for C-style languages.

② A callout behind a line comment for Ruby, Python, Perl, etc.

③ A callout behind a line comment for Clojure.

XML code block with a non-selectable callout

```
[source,xml]
----
<section>
  <title>Section Title</title> <!--1-->
</section>
----
<1> The section title is required.
```

```
  <section>
    <title>Section Title</title> ①
  </section>
```

① The section title is required.

Code block sourced from file

```
[source,ruby]
----
include::app.rb[]
----
```

Code block sourced from file relative to source directory

```
:sourcedir: src/main/java

[source,java]
----
include::{sourcedir}/org/asciidoctor/Asciidoctor.java[]
----
```

Strip leading indentation from source

```
[source,ruby,indent=0]
----
include::lib/document.rb[lines=5..10]
----
```

NOTE
- When `indent` is 0, the leading block indent is stripped (tabs are replaced with 4 spaces).
- When `indent` is > 0, the leading block indent is first stripped (tabs are replaced with 4 spaces), then a block is indented by the number of columns equal to this value.

Code block without delimiters (no blank lines)

```
[source,xml]
<meta name="viewport"
  content="width=device-width, initial-scale=1.0">

This is normal content.
```

```
<meta name="viewport"
   content="width=device-width, initial-scale=1.0">
```

This is normal content.

IMPORTANT	*Enabling the syntax highlighter* Syntax highlighting is enabled by setting the `source-highlighter` attribute in the document header or passed as an argument. `:source-highlighter: pygments` The valid options are `coderay`, `highlightjs`, `prettify`, and `pygments`.

More Delimited Blocks

Sidebar

```
.AsciiDoc history
****
AsciiDoc was first released in Nov 2002 by Stuart Rackham.
It was designed from the start to be a shorthand syntax
for producing professional documents like DocBook and LaTeX.
****
```

AsciiDoc history

AsciiDoc was first released in Nov 2002 by Stuart Rackham. It was designed from the start to be a shorthand syntax for producing professional documents like DocBook and LaTeX.

NOTE	Any block can have a title, positioned above the block. A block title is a line of text that starts with a dot. The dot cannot be followed by a space.

Example

```
.Sample document
====
Here's a sample AsciiDoc document:

[listing]
....
= Title of Document
Doc Writer
:toc:

This guide provides...
....

The document header is useful, but not required.
====
```

Sample document

> Here's a sample AsciiDoc document:
>
> ```
> = Title of Document
> Doc Writer
> :toc:
>
> This guide provides...
> ```
>
> The document header is useful, but not required.

Admonition

```
[NOTE]
====
An admonition block may contain complex content.

.A list
- one
- two
- three

Another paragraph.
====
```

NOTE	An admonition block may contain complex content. *A list* • one • two • three Another paragraph.

TIP	*Admonition and callout icons* Asciidoctor can "draw" icons using Font Awesome and CSS. To use this feature, set the value of the `icons` document attribute to `font`. Asciidoctor will then emit HTML markup that selects an appropriate font character from the Font Awesome font for each admonition block. Icons can also be used inline and styled.

Blockquote

```
[quote, Abraham Lincoln, Address delivered at the dedication of the Cemetery at
Gettysburg]
____
Four score and seven years ago our fathers brought forth
on this continent a new nation...
____

[quote, Albert Einstein]
A person who never made a mistake never tried anything new.


____
A person who never made a mistake never tried anything new.
____

[quote, Charles Lutwidge Dodgson, 'Mathematician and author, also known as
https://en.wikipedia.org/wiki/Lewis_Carroll[Lewis Carroll]']
____
If you don't know where you are going, any road will get you there.
____
```

> Four score and seven years ago our fathers brought forth on this continent a new nation...
>
> — Abraham Lincoln, Address delivered at the dedication of the Cemetery at Gettysburg

> A person who never made a mistake never tried anything new.
>
> — Albert Einstein

> A person who never made a mistake never tried anything new.

> If you don't know where you are going, any road will get you there.
>
> — Charles Lutwidge Dodgson, Mathematician and author, also known as Lewis Carroll

Abbreviated blockquote (Asciidoctor only)

```
"I hold it that a little rebellion now and then is a good thing,
and as necessary in the political world as storms in the physical."
-- Thomas Jefferson, Papers of Thomas Jefferson: Volume 11
```

> I hold it that a little rebellion now and then is a good thing, and as necessary in the political world as storms in the physical.
>
> — Thomas Jefferson, Papers of Thomas Jefferson: Volume 11

Air quotes: the best thing since fenced code blocks (Asciidoctor only)

```
[, James Baldwin]
""
Not everything that is faced can be changed.
But nothing can be changed until it is faced.
""
```

> "" Not everything that is faced can be changed. But nothing can be changed until it is faced. ""

Passthrough

```
++++
<p>
Content in a passthrough block is passed to the output unprocessed.
That means you can include raw HTML, like this embedded Gist:
</p>

<script src="https://gist.github.com/mojavelinux/5333524.js">
</script>
++++
```

Open

```
--
An open block can be an anonymous container,
or it can masquerade as any other block.
--

[source]
--
puts "I'm a source block!"
--
```

An open block can be an anonymous container, or it can masquerade as any other block.

```
puts "I'm a source block!"
```

Custom substitutions

```
:version: 1.5.6.1

[source,xml,subs="verbatim,attributes"]
----
<dependency>
  <groupId>org.asciidoctor</groupId>
  <artifactId>asciidoctor-java-integration</artifactId>
  <version>{version}</version>
</dependency>
----
```

```
<dependency>
    <groupId>org.asciidoctor</groupId>
    <artifactId>asciidoctor-java-integration</artifactId>
    <version>1.5.6.1</version>
</dependency>
```

Block Id, Role and Options

Traditional (longhand) markup method for assigning block id and role

```
[[goals]]
[role="incremental"]
* Goal 1
* Goal 2
```

Shorthand markup method for assigning block id and role (Asciidoctor only)

```
[#goals.incremental]
* Goal 1
* Goal 2
```

TIP
- To specify multiple roles using the shorthand syntax, separate them by dots.
- The order of `id` and `role` values in the shorthand syntax does not matter.

Traditional (longhand) markup method for assigning quoted text anchor (id) and role

```
[[free_the_world]][big goal]_free the world_
```

Shorthand markup method for assigning quoted text anchor (id) and role (Asciidoctor only)

```
[#free_the_world.big.goal]_free the world_
```

Role assigned to text enclosed in backticks

```
[.rolename]`monospace text`
```

Traditional (longhand) markup method for assigning block options

```
[options="header,footer,autowidth"]
|===
|Cell A |Cell B
|===
```

Shorthand markup method for assigning block options (Asciidoctor only)

```
[%header%footer%autowidth]
|===
|Cell A |Cell B
|===
```

Comments

Line

```
// A single-line comment.
```

TIP Single-line comments can be used to divide elements, such as two adjacent lists.

Block

```
////
A multi-line comment.

Notice it's a delimited block.
////
```

Tables

Table with a title, three columns, a header, and two rows of content

```
.Table Title
|===
|Name of Column 1 |Name of Column 2 |Name of Column 3 ①

②
|Cell in column 1, row 1
|Cell in column 2, row 1
|Cell in column 3, row 1

|Cell in column 1, row 2
|Cell in column 2, row 2
|Cell in column 3, row 2
|===
```

Table Title

Name of Column 1	Name of Column 2	Name of Column 3
Cell in column 1, row 1	Cell in column 2, row 1	Cell in column 3, row 1
Cell in column 1, row 2	Cell in column 2, row 2	Cell in column 3, row 2

① Unless the `cols` attribute is specified, the number of columns is equal to the number of cell separator characters on the first (non-blank) line between the block delimiters.

② When a blank line follows the first non-blank line, the cell in the first line get promoted to the table header.

Table with two columns, a header, and two rows of content

```
[%header,cols=2*] ①
|===
|Name of Column 1
|Name of Column 2

|Cell in column 1, row 1
|Cell in column 2, row 1

|Cell in column 1, row 2
|Cell in column 2, row 2
|===
```

Name of Column 1	Name of Column 2
Cell in column 1, row 1	Cell in column 2, row 1
Cell in column 1, row 2	Cell in column 2, row 2

① The * in the `cols` attribute is the repeat operator. It means repeat the column specification across the remaining of columns. In this case, we are repeating the default formatting across 2 columns. When the cells in the header are not defined on a single line, you must use the `cols` attribute to set the number of columns in the table and the `%header` option (or `options=header` attribute) to promote the first row to the table header.

Table with three columns, a header, and two rows of content

```
[cols="1,1,2", options="header"] ①
.Applications
|===
|Name
|Category
|Description

|Firefox
|Browser
|Mozilla Firefox is an open-source web browser.
It's designed for standards compliance,
performance, portability.

|Arquillian
|Testing
|An innovative and highly extensible testing platform.
Empowers developers to easily create real, automated tests.
|===
```

Applications

Name	Category	Description
Firefox	Browser	Mozilla Firefox is an open-source web browser. It's designed for standards compliance, performance, portability.
Arquillian	Testing	An innovative and highly extensible testing platform. Empowers developers to easily create real, automated tests.

① In this example, the `cols` attribute has two functions. It specifies that this table has three columns, and it sets their relative widths.

Table with column containing AsciiDoc content

```
[cols="2,2,5a"]
|===
|Firefox
|Browser
|Mozilla Firefox is an open-source web browser.

It's designed for:

* standards compliance
* performance
* portability

https://www.mozilla.org/en-US/firefox/new[Get Firefox]!
|===
```

Firefox	Browser	Mozilla Firefox is an open-source web browser.
		It's designed for: • standards compliance • performance • portability Get Firefox!

Table from CSV data

```
[%header,format=csv]
|===
Artist,Track,Genre
Baauer,Harlem Shake,Hip Hop
The Lumineers,Ho Hey,Folk Rock
|===
```

Artist	Track	Genre
Baauer	Harlem Shake	Hip Hop
The Lumineers	Ho Hey	Folk Rock

Table from CSV data using shorthand (Asciidoctor only)

```
,===
Artist,Track,Genre

Baauer,Harlem Shake,Hip Hop
,===
```

Artist	Track	Genre
Baauer	Harlem Shake	Hip Hop

Table from CSV data in file

```
|===
include::customers.csv[]
|===
```

Table from DSV data using shorthand (Asciidoctor only)

```
:===
Artist:Track:Genre

Robyn:Indestructable:Dance
:===
```

Artist	Track	Genre
Robyn	Indestructable	Dance

Table with formatted, aligned and merged cells

```
[cols="e,m,^,>s", width="25%"]
|===
|1 >s|2 |3 |4
^|5 2.2+^.^|6 .3+<.>m|7
^|8
|9 2+>|10
|===
```

1	2	3	4
5	6		
8			
9	10	7	

UI Macros

IMPORTANT You **must** set the `experimental` attribute in the document header to enable these macros.

Keyboard shortcuts (inline kbd macro)

```
|===
|Shortcut |Purpose

|kbd:[F11]
|Toggle fullscreen

|kbd:[Ctrl+T]
|Open a new tab

|kbd:[Ctrl+Shift+N]
|New incognito window

|kbd:[\ ]
|Used to escape characters

|kbd:[Ctrl+\]]
|Jump to keyword

|kbd:[Ctrl + +]
|Increase zoom
|===
```

Shortcut	Purpose
`F11`	Toggle fullscreen
`Ctrl`+`T`	Open a new tab
`Ctrl`+`Shift`+`N`	New incognito window
`\`	Used to escape characters
`Ctrl`+`]`	Jump to keyword
`Ctrl`+`+`	Increase zoom

Menu selections (inline menu macro)

```
To save the file, select menu:File[Save].

Select menu:View[Zoom > Reset] to reset the zoom level to the default setting.
```

To save the file, select **File › Save**.

Select **View › Zoom › Reset** to reset the zoom level to the default setting.

Buttons (inline btn macro)

```
Press the btn:[OK] button when you are finished.

Select a file in the file navigator and click btn:[Open].
```

Press the **[OK]** button when you are finished.

Select a file in the file navigator and click **[Open]**.

Attributes and Substitutions

Attribute declaration and usage

```
:url-home: https://asciidoctor.org
:link-docs: https://asciidoctor.org/docs[documentation]
:summary: Asciidoctor is a mature, plain-text document format for \
        writing notes, articles, documentation, books, and more. \
        It's also a text processor & toolchain for translating \
        documents into various output formats (i.e., backends), \
        including HTML, DocBook, PDF and ePub.
:checkedbox: pass:normal[{startsb}&#10004;{endsb}]

Check out {url-home}[Asciidoctor]!

{summary}

Be sure to read the {link-docs} too!

{checkedbox} That's done!
```

Check out Asciidoctor!

Asciidoctor is a mature, plain-text document format for writing notes, articles, documentation, books, and more. It's also a text processor & toolchain for translating documents into various output formats (i.e., backends), including HTML, DocBook, PDF and ePub.

Be sure to read the documentation too!

[] That's done!

Attribute assignment precedence (highest to lowest)

- Attribute passed to the API or CLI that does not end in @

- Attribute defined in the document

- Attribute passed to the API or CLI that ends in @

- Intrinsic attribute value (default values)

TIP To make an attribute value that is passed to the API or CLI have a lower precedence than an assignment in the document, add an @ symbol to the end of the attribute value.

Predefined attributes for character replacements [1][2][3]

Attribute name	Replacement text	Appearance
blank	*nothing*	
empty	*nothing*	
sp	*single space*	

Attribute name	Replacement text	Appearance
nbsp	&#160;	
zwsp[4]	&#8203;	
wj[5]	&#8288;	
apos	&#39;	'
quot	&#34;	"
lsquo	&#8216;	'
rsquo	&#8217;	'
ldquo	&#8220;	"
rdquo	&#8221;	"
deg	&#176;	°
plus	&#43;	+
brvbar	&#166;	¦
vbar	|	|
amp	&	&
lt	<	<
gt	>	>
startsb	[	[
endsb	]	]
caret	^	^
asterisk	*	*
tilde	~	~
backslash	\	\
backtick	`	`
two-colons	::	::
two-semicolons	;;	;;
cpp	C++	C++

[1] Some replacements are Unicode characters, whereas others are numeric character references (e.g., "). These character references are used whenever the use of the Unicode character could interfere with the AsciiDoc syntax or confuse the renderer (i.e., the browser). It's up to the converter to transform the reference into something the renderer understands (something both the man page and PDF converter handle).

[2] Asciidoctor does not prevent you from reassigning predefined attributes. However, it's best to treat them as read-only unless the output format requires the use of a different encoding scheme. These attributes are an effective tool for decoupling content and presentation.

[3] Asciidoctor allows you to use any of the named character references (aka named entities) defined in HTML (e.g., &euro; resolves to €). However, using named character references can cause problems when generating non-HTML output such as PDF because the lookup table needed to resolve these names may not be defined. Our recommendation is avoid using named character references—​with the exception of those defined in XML (i.e., lt, gt, amp, quot and apos). Instead, use numeric character references (e.g., &#8364;).

[4] The Zero Width Space (ZWSP) is a code point in Unicode that shows where a long word can be split if necessary.

[5] The word joiner (WJ) is a code point in Unicode that prevents a line break at its position.

Environment attributes

Attribute	Description	Example Value
asciidoctor	Set if the current processor is Asciidoctor.	
asciidoctor-version	Asciidoctor version.	2.0.10
backend	Backend used to create the output file.	html5
basebackend	The backend value minus any trailing numbers. For example, if the backend is docbook5, the basebackend is docbook.	html
docdate	Last modified date of the source document.[1,2]	2019-01-04
docdatetime	Last modified date and time of the source document.[1,2]	2019-01-04 19:26:06 UTC
docdir	Full path of the directory that contains the source document.	/home/user/docs
docfile	Full path of the source document.	/home/user/docs/userguide.adoc
docfilesuffix	File extension of the source document, including the leading period. *Introduced in 1.5.6.*	.adoc
docname	Root name of the source document (no leading path or file extension).	userguide
doctime	Last modified time of the source document.[1,2]	19:26:06 UTC
doctype	Document type (article, book or manpage).	article
docyear	Year that the document was last modified.[1,2]	2018
embedded	Set if content is being converted to an embeddable document (body only).	
filetype	File extension of the output file name (without leading period).	html
htmlsyntax	Syntax used when generating the HTML output (html or xhtml).	html
localdate	Date when the document was converted.[2]	2019-02-17

Attribute	Description	Example Value
`localdatetime`	Date and time when the document was converted.[2]	`2019-02-17 19:31:05 UTC`
`localtime`	Time when the document was converted.[2]	`19:31:05 UTC`
`localyear`	Year when the document was converted.[2]	`2018`
`outdir`	Full path of the output directory.	`/home/user/docs/dist`
`outfile`	Full path of the output file.	`/home/user/docs/dist/userguide.html`
`outfilesuffix`	File extension of the output file (starting with a period) as determined by the backend (`.html` for `html`, `.xml` for `docbook`, etc.). (The value is not updated to match the file extension of the output file when one is specified explicitly). *Safe to modify.*	`.html`
`safe-mode-level`	Numeric value of the safe mode setting. (UNSAFE=0, SAFE=10, SERVER=10, SECURE=20).	`20`
`safe-mode-name`	Textual value of the safe mode setting.	`SERVER`
`safe-mode-unsafe`	Set if the safe mode is UNSAFE.	
`safe-mode-safe`	Set if the safe mode is SAFE.	
`safe-mode-server`	Set if the safe mode is SERVER.	
`safe-mode-secure`	Set if the safe mode is SECURE.	
`user-home`	Home directory of the current user. Resolves to `.` if the safe mode is SERVER or greater.	`/home/user`

[1] Only reflects the last modified time of the source document file. It does not consider the last modified time of files which are included.

[2] If the SOURCE_DATE_EPOCH environment variable is set, the value assigned to this attribute is built from a UTC date object that corresponds to the timestamp (as an integer) stored in that environment variable. This override offers one way to make the conversion reproducible. See https://reproducible-builds.org/specs/source-date-epoch/ for more information about the SOURCE_DATE_EPOCH environment variable. Otherwise, the date is expressed in the local time zone, which is reported as a time zone offset (e.g., `-0600`) or UTC if the time zone offset is 0). To force the use of UTC, set the `TZ=UTC` environment variable when invoking Asciidoctor.

Named substitutions

none

    Disables substitutions

normal

    Performs all substitutions except for callouts

verbatim

Replaces special characters and processes callouts

specialchars, **specialcharacters**

Replaces <, >, and & with their corresponding entities

quotes

Applies text formatting

attributes

Replaces attribute references

replacements

Substitutes textual and character reference replacements

macros

Processes macros

post_replacements

Replaces the line break character (+)

Counter attributes

```
[caption=""]
.Parts{counter2:index:0}
|===
|Part Id |Description

|PX-{counter:index}
|Description of PX-{index}

|PX-{counter:index}
|Description of PX-{index}
|===
```

Parts

Part Id	Description
PX-1	Description of PX-1
PX-2	Description of PX-2

Text Replacement

Textual symbol replacements

Name	Syntax	Unicode Replacement	Rendered	Notes
Copyright	`(C)`	`&#169;`	©	
Registered	`(R)`	`&#174;`	®	
Trademark	`(TM)`	`&#8482;`	™	
Em dash	`--`	`&#8212;`	—	Only replaced if between two word characters, between a word character and a line boundary, or flanked by spaces. When flanked by space characters (e.g., `a -- b`), the normal spaces are replaced by thin spaces ( ).
Ellipsis	`...`	`&#8230;`	…	
Single right arrow	`->`	`&#8594;`	→	
Double right arrow	`=>`	`&#8658;`	⇒	
Single left arrow	`<-`	`&#8592;`	←	
Double left arrow	`<=`	`&#8656;`	⇐	
Typographic apostrophe	`Sam's`	`Sam&#8217;s`	Sam's	The typewriter apostrophe is replaced with the typographic (aka curly) apostrophe.

TIP Any named, numeric or hexadecimal XML character reference is supported.

Escaping Text

Backslash

```
\*Stars* is not rendered as bold text.
The asterisks around the word are preserved.

\{author} is not resolved to the author name.
The curly brackets around the word are preserved.

`A\--Z` connects A to Z in monospace using two dashes.
The dashes are not replaced by an em dash.

\[[Word]] is not interpreted as an anchor.
The double brackets around the word are preserved.

[\[[Word]]] is not interpreted as a bibliography anchor.
The triple brackets around the word are preserved.

In these cases, the backslash character is automatically removed.
```

Stars is not rendered as bold text. The asterisks around the word are preserved.

{author} is not resolved to the author name. The curly brackets around the word are preserved.

`A--Z` connects A to Z in monospace using two dashes. The dashes are not replaced by an em dash.

[[Word]] is not interpreted as an anchor. The double brackets around the word are preserved.

[[[Word]]] is not interpreted as a bibliography anchor. The triple brackets around the word are preserved.

In these cases, the backslash character is automatically removed.

Passthrough ("plus for passthrough")

```
Text between + characters, such as +/user/{name}+, is not substituted.
However, special characters like +<+ and +>+ are still escaped.

You can write `C++{plus}{plus}++` as `+{cpp}+`.
```

Text between + characters, such as /user/{name}, is not substituted. However, special characters like < and > are still escaped.

You can write `C{plus}{plus}` as `{cpp}`.

Raw (triple plus and inline pass macro)

```
+++<u>underline me</u>+++ is underlined.

pass:[<u>underline me</u>] is also underlined.
```

<u>underline me</u> is underlined.

<u>underline me</u> is also underlined.

Table of Contents (ToC)

Document with ToC

```
= AsciiDoc Writer's Guide
Doc Writer <doc.writer@asciidoctor.org>
v1.0, 2013-08-01
:toc:
```

Document with ToC positioned on the right

```
= AsciiDoc Writer's Guide
Doc Writer <doc.writer@asciidoctor.org>
v1.0, 2014-08-01
:toc: right
```

TIP The ToC title, levels, and positioning can be customized.

Bibliography

Bibliography with inbound references

```
_The Pragmatic Programmer_ <<pp>> should be required reading for all developers.
To learn all about design patterns, refer to the book by the "`Gang of Four`" <<gof>>.

[bibliography]
== References

- [[[pp]]] Andy Hunt & Dave Thomas. The Pragmatic Programmer:
  From Journeyman to Master. Addison-Wesley. 1999.
- [[[gof,2]]] Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides. Design
Patterns:
  Elements of Reusable Object-Oriented Software. Addison-Wesley. 1994.
```

The Pragmatic Programmer [pp] should be required reading for all developers. To learn all about design patterns, refer to the book by the "Gang of Four" [2].

References

- [pp] Andy Hunt & Dave Thomas. The Pragmatic Programmer: From Journeyman to Master. Addison-Wesley. 1999.

- [2] Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. 1994.

Footnotes

Normal and reusable footnotes

```
A statement.footnote:[Clarification about this statement.]

A bold statement!footnote:disclaimer[Opinions are my own.]

Another bold statement.footnote:disclaimer[]
```

A statement.[3]

A bold statement![4]

Another bold statement.[2]

Markdown Compatibility

Markdown compatible syntax is only available when using Asciidoctor.

Markdown-style headings

```
# Document Title (Level 0)

## Section Level 1

### Section Level 2

#### Section Level 3

##### Section Level 4

###### Section Level 5
```

Document Title (Level 0)

Section Level 1

Section Level 2

Section Level 3

Section Level 4

Section Level 5

Fenced code block with syntax highlighting

````
```ruby
require 'sinatra'

get '/hi' do
  "Hello World!"
end
```
````

```
require 'sinatra'

get '/hi' do
  "Hello World!"
end
```

Markdown-style blockquote

```
> I hold it that a little rebellion now and then is a good thing,
> and as necessary in the political world as storms in the physical.
> -- Thomas Jefferson, Papers of Thomas Jefferson: Volume 11
```

> I hold it that a little rebellion now and then is a good thing, and as necessary in the political world as storms in the physical.
>
> — Thomas Jefferson, Papers of Thomas Jefferson: Volume 11

Markdown-style blockquote with block content

```
> > What's new?
>
> I've got Markdown in my AsciiDoc!
>
> > Like what?
>
> * Blockquotes
> * Headings
> * Fenced code blocks
>
> > Is there more?
>
> Yep. AsciiDoc and Markdown share a lot of common syntax already.
```

> > What's new?
>
> I've got Markdown in my AsciiDoc!
>
> > Like what?
>
> * Blockquotes
> * Headings
> * Fenced code blocks
>
> > Is there more?
>
> Yep. AsciiDoc and Markdown share a lot of common syntax already.

Markdown-style horizontal rules

```
---

- - -

***

* * *
```

User Manual and Help

To learn more about Asciidoctor and its capabilities, check out the other Asciidoctor guides and its User Manual. Also, don't forget to join the Asciidoctor mailing list, where you can ask questions and leave comments.