

# AsciiDoc Syntax Quick Reference

Dan Allen

Sarah White

Version 1.0.4, 2014-08-08

## Table of Contents

Paragraphs

Formatted Text

Document Header

Section Titles (Headers)

Include Files

Horizontal Rules and Page Breaks

Lists

Links

Images

Videos

Source Code

More Delimited Blocks

Block Id, Role and Options

Comments

Tables

UI Macros

Attributes and Substitutions

Text Replacement

Escaping Text

Table of Contents (ToC)

Bibliography

Footnotes

Markdown Compatibility (AsciiDoctor only)

User Manual and Help

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.



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

Most features labeled *Asciidoctor only* can be made available to the legacy AsciiDoc Python processor by using [this AsciiDoc configuration file](https://github.com/asciidoctor/asciidoctor/blob/master/compat/asciidoc.conf) (<https://github.com/asciidoctor/asciidoctor/blob/master/compat/asciidoc.conf>) provided by the Asciidoctor project.

# 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.

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.

## *Line breaks*

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.

## *Literal*

A normal paragraph.

A sequence of lines that begin with at least one space is a literal paragraph.  
Literal paragraphs are treated as preformatted text.  
The text is shown in a fixed-width font  
and endlines are preserved.

Another normal paragraph.

A normal paragraph.

A sequence of lines that begin with at least one space is a literal paragraph. Literal paragraphs are treated as preformatted text. The text is shown in a fixed-width font and endlines 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...



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:



Pro tip...



Don't forget...



Watch out for...



Ensure that...



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).



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 \(http://asciidoctor.org/docs/user-manual/#paragraph\)](http://asciidoctor.org/docs/user-manual/#paragraph)
- [Literal Text and Blocks \(http://asciidoctor.org/docs/user-manual/#literal-text-and-blocks\)](http://asciidoctor.org/docs/user-manual/#literal-text-and-blocks)
- [Admonitions \(http://asciidoctor.org/docs/user-manual/#admonition\)](http://asciidoctor.org/docs/user-manual/#admonition)

# Formatted Text

## *Bold, Italic, and Monospace*

`_italic phrase_`

`__i__talic le__tt__ers`

`*bold phrase*`

`**b**old le**tt**ers`

`*_bold italic phrase_*`

`**__b__**old italic le**__tt__**ers`

``monospace phrase` and le``tt``ers`

``_monospace italic phrase_` and le``__tt__``ers`

``*monospace bold phrase*` and le``**tt**``ers`

``*_monospace bold italic phrase_*` and le``**__tt__**``ers`

*italic phrase*

*italic letters*

**bold phrase**

**bold letters**

***bold italic phrase***

***bold italic letters***

`monospace phrase` and `le tt ers`

`monospace italic phrase` and `le tt ers`

`monospace bold phrase` and `le tt ers`

`monospace bold italic phrase` and `le tt ers`

## Custom Styling

Werewolves are #allergic to cinnamon#.

Do werewolves believe in [small]#small print#?

[big]##0##nce upon an infinite loop.

Werewolves are allergic to cinnamon.

Do werewolves believe in small print?

Once upon an infinite loop.

## Superscript and Subscript

<sup>super</sup> phrase

<sub>sub</sub> phrase

<sup>super</sup>script phrase

<sub>sub</sub>script phrase

## Curved Quotation Marks and Apostrophes

``single curved quotes``

`"double 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.`

‘single curved quotes’

“double 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 \(http://asciidoctor.org/docs/user-manual/#bold-and-italic\)](http://asciidoctor.org/docs/user-manual/#bold-and-italic)
- [Quotation Marks and Apostrophes \(http://asciidoctor.org/docs/user-manual/#quotation-marks-and-apostrophes\)](http://asciidoctor.org/docs/user-manual/#quotation-marks-and-apostrophes)
- [Subscript and Superscript \(http://asciidoctor.org/docs/user-manual/#subscript-and-superscript\)](http://asciidoctor.org/docs/user-manual/#subscript-and-superscript)
- [Monospace Formatting \(http://asciidoctor.org/docs/user-manual/#monospace\)](http://asciidoctor.org/docs/user-manual/#monospace)
- [Custom Styling with Attributes \(http://asciidoctor.org/docs/user-manual/#custom-styling-with-attributes\)](http://asciidoctor.org/docs/user-manual/#custom-styling-with-attributes)
- [Passthrough Macros \(http://asciidoctor.org/docs/user-manual/#passthrough-macros\)](http://asciidoctor.org/docs/user-manual/#passthrough-macros)

# Document Header



A header is optional.



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...
```



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...
```



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, 2014-01-01
:toc:
:imagesdir: assets/images
:homepage: http://asciidoctor.org
```

My document provides...

# Section Titles (Headers)

## *Article doctype*

```
= 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



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.



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

## *Book doctype*

```
= 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 links.



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[]
```



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*

```
:asciidoctor-source: https://raw.githubusercontent.com/asciidoctor/asciidoctor/master  
  
include::{asciidoctor-source}/README.adoc[]
```



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.

# Horizontal Rules and Page Breaks

## Horizontal rule

...

\_\_\_\_\_

## Page break

<<<



# Lists

## *Unordered, basic*

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

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



Blank lines are required before and after a list.



You can force two lists apart with a line comment, as the previous example demonstrates. The text in the comment, ( ^ ), is optional, but serves as a hint to other authors that this line serves as an “end of list” marker.

## *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



The unordered list marker can be changed using [block styles](http://asciidoctor.org/docs/user-manual/#custom-markers) (<http://asciidoctor.org/docs/user-manual/#custom-markers>).

### *Checklist*

- [\*] checked
- [x] also checked
- [ ] not checked
- normal list item

- ☒ checked
- ☒ also checked
- ☐ not checked
- normal list item



Checklists can use [font-based icons](http://asciidoctor.org/docs/user-manual/#checklist) and be interactive (<http://asciidoctor.org/docs/user-manual/#checklist>).

### *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



For ordered lists, AsciiDoctor supports numeration styles (<http://asciidoctor.org/docs/user-manual/#numeration-styles>) such as `lowergreek` and `decimal-leading-zero`.

### *Labeled, single-line*

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

**first term**

definition of first term

**section term**

definition of second term

*Labeled, multi-line*

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

**first term**

definition of first term

**section 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
  - o Desktop
2. Ubuntu
  - o Desktop
  - o Server

### BSD

1. FreeBSD
2. NetBSD

## Cloud Providers

### PaaS

1. OpenShift
2. CloudBees

### IaaS

1. Amazon EC2

2. Rackspace



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*

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

`http://asciidoctor.org[Asciidoctor]`

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

<http://asciidoctor.org> (<http://asciidoctor.org>) - automatic!

[Asciidoctor](http://asciidoctor.org) (<http://asciidoctor.org>)

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

## *Relative*

`link:index.html[Docs]`

[Docs](#)

## *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/#asciidoctor`

`devel@discuss.arquillian.org`

`Discuss Arquillian`

`Subscribe`



<irc://irc.freenode.org/#asciidoctor>

### *Link with attributes (Asciidoctor only)*

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

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

```
http://search.example.com["Google, Yahoo, Bing^", role="teal"]
```

Discuss Asciidoctor (<http://discuss.asciidoctor.org>)

Discuss Asciidoctor (<http://discuss.asciidoctor.org>)

Google, Yahoo, Bing (<http://search.example.com>)



Links with attributes (including the subject and body segments on mailto links) are a feature unique to Asciidoctor. To enable them, you must set the `linkattrs` attribute on the document. When they are enabled, you must quote the link text if it contains a comma.

### *Inline anchors*

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

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

Inline anchors make arbitrary content referenceable.

Use a cross reference to link to this location.

### *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

## *Block*

```
image::sunset.jpg[]
```

```
image::sunset.jpg[Sunset]
```

```
[[img-sunset]]
```

```
image::sunset.jpg[caption="Figure 1: ", title="A mountain sunset", alt="Sunset",  
width="300", height="200", link="http://www.flickr.com/photos/javh/5448336655"]
```

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





(<http://www.flickr.com/photos/javh/5448336655>)

*Figure 1: A mountain sunset*



Images are resolved relative to the value of the `imagesdir` document attribute (<http://asciidoctor.org/docs/user-manual/#set-the-images-directory>), which defaults to an empty value. The `imagesdir` attribute can be an absolute path, relative path or base URL. If the image target is a URL or absolute path, the `imagesdir` prefix is *not* added.



You should use the `imagesdir` attribute to avoid hard coding the shared path to your images in every image macro.

## Image macro using positioning role

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

What a beautiful sunset!




There are a variety of attributes available to position and frame images (<http://asciidoctor.org/docs/user-manual/#put-images-in-their-place>).

## 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.

## Embedded

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



When the `data-uri` attribute is set, all images in the document—including admonition icons—are embedded into the document as data URIs ([https://developer.mozilla.org/en-US/docs/data\\_URIs](https://developer.mozilla.org/en-US/docs/data_URIs)).



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]
```



You can control the video settings using [additional attributes and options](http://asciidoctor.org/docs/user-manual/#video) (<http://asciidoctor.org/docs/user-manual/#video>) on the macro.

# Source Code

## *Inline*

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

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

## *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 (0.1.4)

PLATFORMS
  ruby

DEPENDENCIES
  asciidoctor (~> 0.1.4)
----
```

### *Gemfile.lock*

```
GEM
  remote: https://rubygems.org/
  specs:
    asciidoctor (0.1.4)

PLATFORMS
  ruby

DEPENDENCIES
  asciidoctor (~> 0.1.4)
```

### *Code block with title and syntax highlighting*

```
[source,ruby]
.app.rb
----
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> Content for response
```

```
require 'sinatra' ❶

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

- ❶ Library import
- ❷ URL mapping
- ❸ Content for response

### *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 ❶  
line of code ❷  
line of code ❸
```

- ❶ A callout behind a line comment for C-style languages.
- ❷ A callout behind a line comment for Ruby, Python, Perl, etc.
- ❸ 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> ❶  
</section>
```

- ❶ 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]
```

```
----
```



- 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.

### *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 for 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.



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.
====
```



An admonition block may contain complex content.

#### *A list*

- one
- two
- three

Another paragraph.

### *Admonition and callout icons*

AsciiDoctor can “draw” icons using [Font Awesome](https://fontawesome.github.io/Font-Awesome/) (<https://fontawesome.github.io/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](http://asciidoctor.org/docs/user-manual/#inline-icons) (<http://asciidoctor.org/docs/user-manual/#inline-icons>) and [styled](http://asciidoctor.org/docs/user-manual/#size-rotate-and-flip) (<http://asciidoctor.org/docs/user-manual/#size-rotate-and-flip>).

### *Blockquote*

[quote, Abraham Lincoln, Soldiers' National Cemetery Dedication]

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.

“Four score and seven years ago our fathers brought forth on this  
continent a new nation...

— Abraham Lincoln  
Soldiers' National Cemetery Dedication

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

— Albert Einstein

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

### *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)*

```
[, Richard M. Nixon]
""
When the President does it, that means that it's not illegal.
""
```

“*When the President does it, that means that it's not illegal.*

— Richard M. Nixon

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

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



```

1  # The Greeter class
2  class Greeter
3      def initialize(name)
4          @name = name.capitalize
5      end
6
7      def salute
8          puts "Hello #{@name}!"
9      end
10 end
11
12 # Create a new object
13 g = Greeter.new("world")
14
15 # Output "Hello World!"
16 g.salute

```

greeter.rb ([https://gist.github.com/mojavelinux/view\\_raw](https://gist.github.com/mojavelinux/view_raw) (<https://gist.github.com/mojavelinux/5333524/raw/greeter.rb>)  
/5333524#file-greeter-rb) hosted with ♥ by [GitHub](https://github.com) (<https://github.com>)

## 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: 0.1.4
```

```
[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>0.1.4</version>
```

```
</dependency>
```

# Block Id, Role and Options

## *Traditional 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
```



- 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 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 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.
```



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 ❶
❷
|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
|===
```

- ❶ Unless the `cols` attribute is specified, the number of columns is equal to the number of vertical bars on the first non-blank line inside the block delimiters.
- ❷ When a blank line follows a single line of column titles, the column titles row will be styled as a header row by default.

*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

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

```
[cols="2*", options="header"] ❶
|===
|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
|===
```

The `*` in the `cols` attribute is the repeat operator. It means repeat the column specification for the remainder of columns. In this case, it means to repeat the default ❶ formatting across 4 columns. When the header row is not defined on a single line, you must use the `cols` attribute to set the number of columns and `options` attributes to make the first row a header.

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

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

```
[cols="1,1,2", options="header"] ❶
.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.
|==
```

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

### *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.

*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

```
http://getfirefox.com[Get Firefox]!
|===
```

Firefox	Browser	Mozilla Firefox is an open-source web browser.  It's designed for: <ul style="list-style-type: none"><li>• standards compliance</li><li>• performance</li><li>• portability</li></ul> <u><a href="http://getfirefox.com">Get Firefox (http://getfirefox.com)</a></u>
---------	---------	---

### *Table from CSV data*

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

Artist	Track	Genre
Baauer	Harlem Shake	Hip Hop

Artist	Track	Genre
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		

# UI Macros



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

Shortcut	Purpose
<div>F11</div>	Toggle fullscreen
<div>Ctrl + T</div>	Open a new tab
<div>Ctrl + Shift + N</div>	New incognito window
<div>Ctrl + +</div>	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*

```
:homepage: http://asciidoctor.org
:docslink: http://asciidoctor.org/docs[Asciidoctor's Docs]
:description: 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[+[#10004;]+]

Check out {homepage}[Asciidoctor]!

{description}

Check out {docslink} too!

{checkedbox} That's done!
```

Check out Asciidoctor (<http://asciidoctor.org>)!

This guide is a quick reference for the common AsciiDoc document and text formatting markup.

Check out Asciidoctor's Docs (<http://asciidoctor.org/docs>) 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)



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.

### *Built-in literal attributes*

Attribute reference	Replacement	Rendered
{lt}	<	<
{gt}	>	>
{amp}	&	&
{startsb}	[	[
{endsb}	]	]
{vbar}		
{caret}	^	^
{asterisk}	*	*
{tilde}	~	~
{apostrophe}	'	'
{backslash}	\	\
{backtick}	`	`
{two-colons}	::	::
{two-semicolons}	:::	:::

### *Built-in entity attributes*

Attribute reference	Replacement	Rendered
{empty}	<i>nothing</i>	
{sp}, {space}	<i>single space</i>	
{nbsp}	&#160;	
{zwsp}	&#8203;	
{wj}	&#8288;	
{apos}	&#39;	'
{quot}	&#34;	"
{lsquo}	&#8216;	‘
{rsquo}	&#8217;	’
{ldquo}	&#8220;	“
{rdquo}	&#8221;	”
{deg}	&#176;	°
{plus}	&#43;	+
{brvbar}	&#166;	

### *Built-in data attributes*

Attribute	Description
asciidoctor	Calls the processor
asciidoctor-version	Version of the processor



Attribute	Description
backend	Backend used to render document
docdate	Last modified date
docdatetime	Last modified date and time
docdir	Name of document directory
docfile	Name of document file
doctime	Last modified time
doctitle	The title of the document
doctype	Document's doctype (e.g., article)
localdate	Local date when rendered
localdatetime	Local date and time when rendered
localtime	Local time when rendered

### *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;	—	When space is detected on either side of the em dash, the thin space numeric character entity (&#8201;) is also substituted into the document.
ellipses	...	&#8230;	...	
right single arrow	->	&#8594;	→	
right double arrow	=>	&#8658;	⇒	
left single arrow	<-	&#8592;	←	
left double arrow	<=	&#8656;	⇐	
apostrophe	Sam ' s	Sam&#8217;s	Sam's	The vertical form apostrophe is replaced with the curved form apostrophe.



Any named, numeric or hex [XML entity reference \(https://en.wikipedia.org/wiki/List\\_of\\_XML\\_and\\_HTML\\_character\\_entity\\_references\)](https://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references) 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.

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.

The backslash character is automatically removed.

## *Triple plus and inline passthrough macro*

`+++<u>underline me</u>+++` renders as underlined text.

`pass:[<u>underline me</u>]` also renders as underlined text.

underline me renders as underlined text.

underline me also renders as underlined text.

# 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
```



The ToC title, levels, and positioning (<http://asciidoctor.org/docs/user-manual/#user-toc>) can be customized.

# Bibliography

## References

\_The Pragmatic Programmer\_ <<prag>> should be required reading for all developers.

[bibliography]

- [[[prag]]] Andy Hunt & Dave Thomas. The Pragmatic Programmer: From Journeyman to Master. Addison-Wesley. 1999.
- [[[seam]]] Dan Allen. Seam in Action. Manning Publications. 2008.

*The Pragmatic Programmer* [prag] should be required reading for all developers.

- [prag] Andy Hunt & Dave Thomas. The Pragmatic Programmer: From Journeyman to Master. Addison-Wesley. 1999.
- [seam] Dan Allen. Seam in Action. Manning Publications. 2008.

# Footnotes

## *Normal and reusable footnotes*

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

A bold statement.footnoteref:[disclaimer,Opinions are my own.]

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

A statement.<sup>[1]</sup>

A bold statement.<sup>[2]</sup>

Another bold statement.<sup>[2]</sup>

---

**1** Clarification about this statement.

**2** Opinions are my own.

# Markdown Compatibility (Asciidoctor only)

## *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](http://asciidoctor.org/docs) (<http://asciidoctor.org/docs>) and its [User Manual](http://asciidoctor.org/docs/user-manual) (<http://asciidoctor.org/docs/user-manual>). Also, don't forget to join the [AsciiDoctor mailing list](https://discuss.asciidoctor.org) (<https://discuss.asciidoctor.org>), where you can ask questions and leave comments.

Version 1.0.4

Last updated 2014-08-11 19:40:36 MDT