

AsciiDoc Syntax

This is the Book Subtitle

Urs Roesch

Version 1.2.3, 2020-11-11: Syntax Test

Table of Contents

Paragraphs	2
Text Formatting	3
Attributes	4
Links	5
Anchors	6
Lists	7
Unordered	7
Ordered	7
Checklist	7
Callout	7
Description	8
Document Structure	9
Header	9
Sections	9
Includes	9
Blocks	10
Tables	11
Multimedia	12
Breaks	15

A brief reference of the most commonly used AsciiDoc syntax. You can find the full documentation for the AsciiDoc syntax at <https://asciidoctor.org/docs>.

Paragraphs

A normal paragraph. Line breaks are not preserved.

A blank line separates paragraphs.

This paragraph carries the **hardbreaks** option.

Notice how line breaks are now preserved.

An indented (literal) paragraph disables text formatting, preserves spaces and line breaks, and is displayed in a monospaced font.

A style, ID, and/or role gives a paragraph (or block) special meaning, like this sidebar.



An admonition paragraph, like this note, grabs the reader's attention.



Convert this document using the **asciidoctor** command to see the output produced from it.

Text Formatting

Constrained (applied at word boundaries)

strong importance (aka bold)

stress emphasis (aka italic)

monospaced (aka typewriter text)

“double” and ‘single’ typographic quotes

passthrough text (substitutions disabled)

literal text (monospaced with substitutions disabled)

Unconstrained (applied anywhere)

Create+**Read**+**Update**+Delete

fanfreakintastic

monoculture


Replacements

A long time ago in a galaxy far, far away...

© 1976 Arty Artisan

I believe I shall—no, actually I won't.

Macros

The European  is blue & contains ***** arranged in a .

The -> operator is often referred to as the stabby lambda.

Since ++ has strong priority in AsciiDoc, you can rewrite C++ ⇒ C++.

sqrt(4) = 2

Attributes

```
// define attributes in the document header; must be flush with left margin
:name: value
```

You can download and install Asciidoctor 2.0.10 from <https://rubygems.org/gems/asciidoctor>. C++ is not required, only Ruby. Use a leading backslash to output a word enclosed in curly braces, like {name}.

Links

[A webpage](#)

[A local file](#)

[A sibling document](#)

[Email to say hello!](#)

Anchors

A paragraph (or any block) with an anchor (aka ID) and reftext.

See [reference text](#) or [optional text of internal link](#).

[Jumps to anchor in another document](#).

This paragraph has a footnote.^[1]

[1] This is the text of the footnote.

Lists

Unordered

- level 1
 - level 2
 - level 3
 - level 4
 - etc.
- back at level 1

Attach a block or paragraph to a list item using a list continuation (which you can enclose in an open block).

Some Authors

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

Ordered

1. Step 1
2. Step 2
 - a. Step 2a
 - b. Step 2b
3. Step 3

Remember your Roman numerals?

- I. is one
- II. is two
- III. is three

Checklist

- ☒ checked
- ☐ not checked

Callout

```
puts 'Hello, World!' ❶
```

① Prints **Hello, World!** to the console.

Description

first term

description of first term

second term

description of second term

Document Structure

Header

```
// header must be flush with left margin
= Document Title
Author Name <author@example.org>
v1.0, 2019-01-01
```

Sections

```
// must be flush with left margin
= Document Title (Level 0)
== Level 1
=== Level 2
==== Level 3
===== Level 4
===== Level 5
== Back at Level 1
```

Includes

```
// must be flush with left margin
include::basics.adoc[]
```

```
// define -a allow-uri-read to allow content to be read from URI
include::https://example.org/installation.adoc[]
```

Blocks

open - a general-purpose content wrapper; useful for enclosing content to attach to a list item



admonition - a notice for the reader, ranging in severity from a tip to an alert

example - a demonstration of the concept being documented

Example 1. Toggle Me

collapsible - these details are revealed by clicking the title

sidebar - auxiliary content that can be read independently of the main content

```
literal - an exhibit that features program output
```

```
listing - an exhibit that features program input, source code, or the contents of  
a file
```

```
source - a listing that is embellished with (colorized) syntax highlighting
```

```
fenced code - a shorthand syntax for the source block
```

quote - a quotation or excerpt; attribution with title of source are optional

— attribution, citetitle

verse - a literary excerpt, often a poem; attribution with title of source are optional

— attribution, citetitle

```
pass - content passed directly to the output document; often raw HTML
```

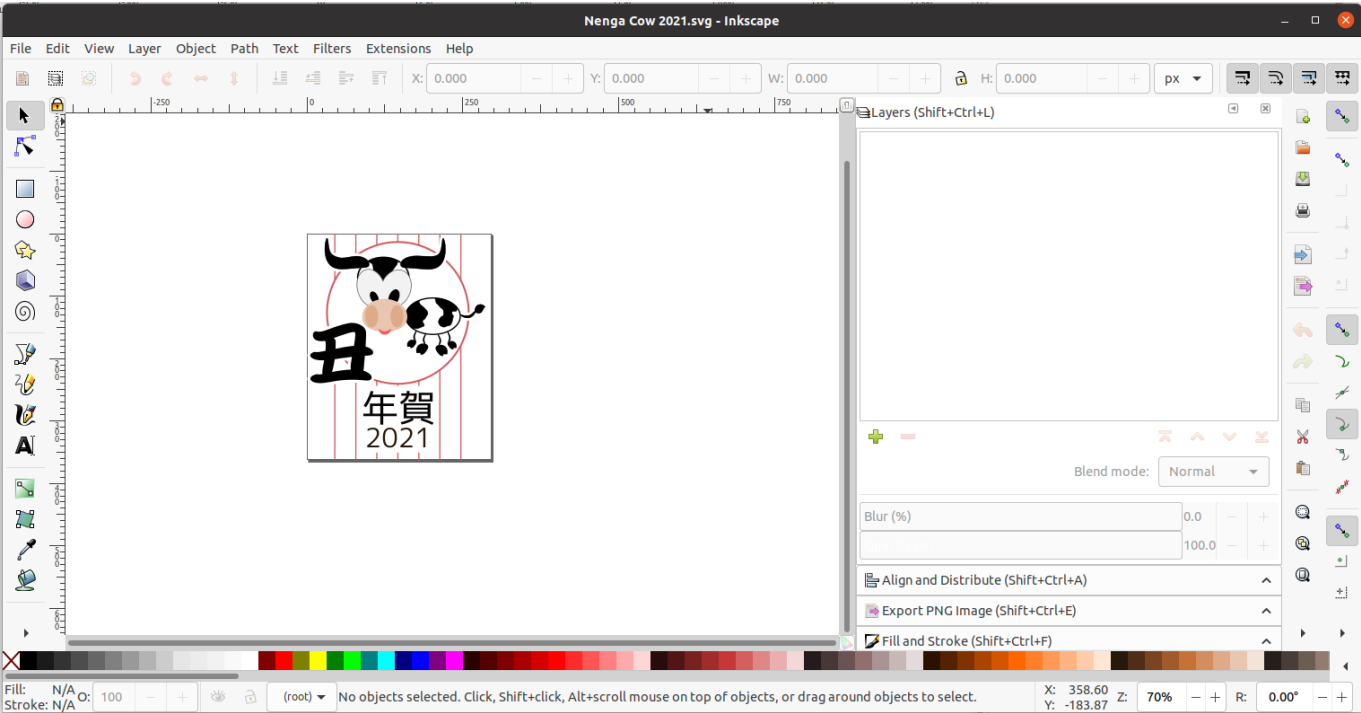
```
x = y^2
```

Tables

Table 1. Table Attributes

Attribute Name	Values
options	header,footer,autowidth
cols	colspec[;colspec;...]
grid	all cols rows none
frame	all sides topbot none
stripes	all even odd none
width	(0%..100%)
format	psv csv dsv

Multimedia





▶ [movie.mp4](#) (*video*)

▶ <https://www.youtube.com/watch?v=aHjpOzsQ9YI> (*YouTube video*)

[] | *vimeo*

